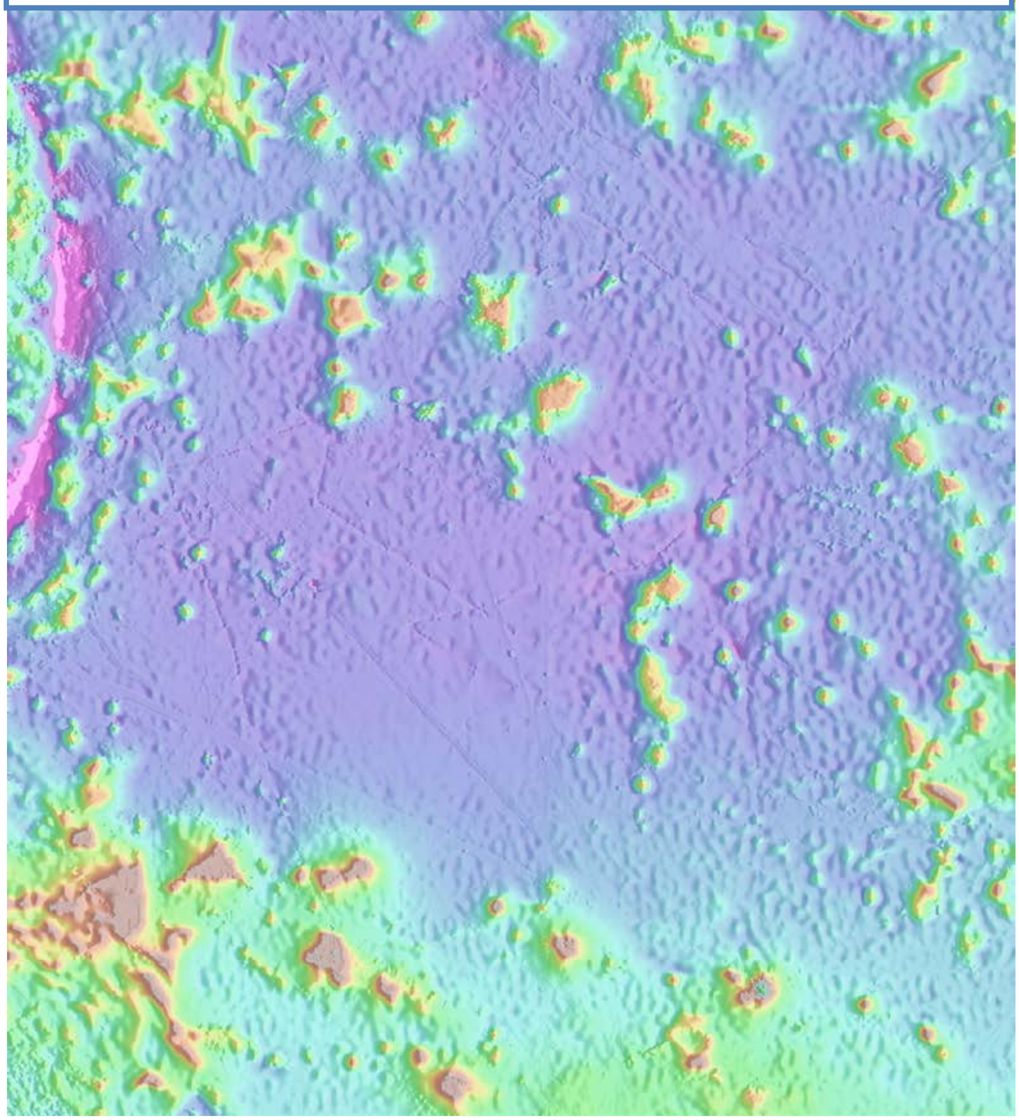
United Nations Educational, Scientific and Cultural Organization

Intergovernmental Oceanographic Commission

The IHO-IOC GEBCO Cook Book



November 2013

IHO Publication B-11
IOC Manuals and Guides, 63

Contributors

Directorate of Hydrography and Navigation (DHN)
Center of Hydrography and Navigation (CHN)
Brazilian Navy, Brazil
http://www.mar.mil.br/dhn/

British Oceanographic Data Centre (BODC), UK
http://www.bodc.ac.uk

Canadian Hydrographic Service, Canada
http://www.charts.gc.ca

CARIS Geospatial Software Solutions, Canada
http://www.caris.com

Center for Coastal & Ocean Mapping
Joint Hydrographic Center
University of New Hampshire, USA
http://ccom.unh.edu

Centro de Investigáciones Oceanograficas e Hidrográficas (CIOH), Columbia
http://www.cioh.org.co/

Cooperative Institute for Research in Environmental Sciences (CIRES)
University of Colorado at Boulder
http://cires.colorado.edu

International Hydrographic Bureau, Monaco
http://www.iho.int

Geological Survey of Israel, Israel
http://www.gsi.gov.il

National Oceanography Centre, UK
http://noc.ac.uk/

Naval Research Laboratory, USA
http://www.nrl.navy.mil

Nigerian Navy Hydrographic Office
Lagos, Nigeria
http://nnhomsi.com

NOAA Laboratory for Satellite Altimetry, USA
http://ibis.grdl.noaa.gov

NOAA National Geophysical Data Center, USA
http://www.ngdc.noaa.gov

Geological Institute of Russian Academy of Sciences
Laboratory of Geomorphology and Ocean Floor Tectonics, Russia
http://atlantic.ginras.ru/index.html

Scripps Institution of Oceanography
University of California at San Diego
http://www.sio.ucsd.edu

Service Hydrographique et Océanographique de la Marine
Brest, France
http://www.shom.fr

Department of Geological Sciences
Stockholm University, Sweden
http://www.su.se

Table of Contents

List of Figures

List of Tables

Preface

At its 26th meeting in Brest, France, in October 2010, the GEBCO Guiding Committee (GGC) approved a proposal from its Technical Sub-Committee on Ocean Mapping (TSCOM) that they prepare a publication to be called the "GEBCO Cook Book" to provide an educational resource for a wide range of persons interested in preparing gridded datasets from bathymetric data.

This publication includes inputs from a wide range of individuals and organizations all of whom are experts in their respective fields, as set out in more detail in the opening pages and in Chapter 1. It is intended that this will be a living document that will be maintained on a continuous basis as new or amended techniques and software become available. Future updates will be prepared by the Chief Editor under the direction of TSCOM and endorsed by the GGC. Dr. Karen Marks of NOAA has acted as the Chief Editor on behalf of TSCOM for the preparation of the initial text.

The initial text was endorsed by the GGC at its 28th meeting in San Diego, USA, in October 2011, adopted by the IHO as announced in CL 47/2012 in May 2012, and approved by the IOC Executive Council in October 2012.

The IHO-IOC GEBCO Cook Book, IHO Publication B-11, is available for free download via links from the GEBCO web site (http://www.gebco.net) and the IHO Publication download web page http://www.iho.int/iho_pubs/IHO_Download.htm. This publication is also available as IOC Manuals and Guides, 63 on the IOC website http://www.ioc-unesco.org/documents.

Chapter 1.0 Introduction

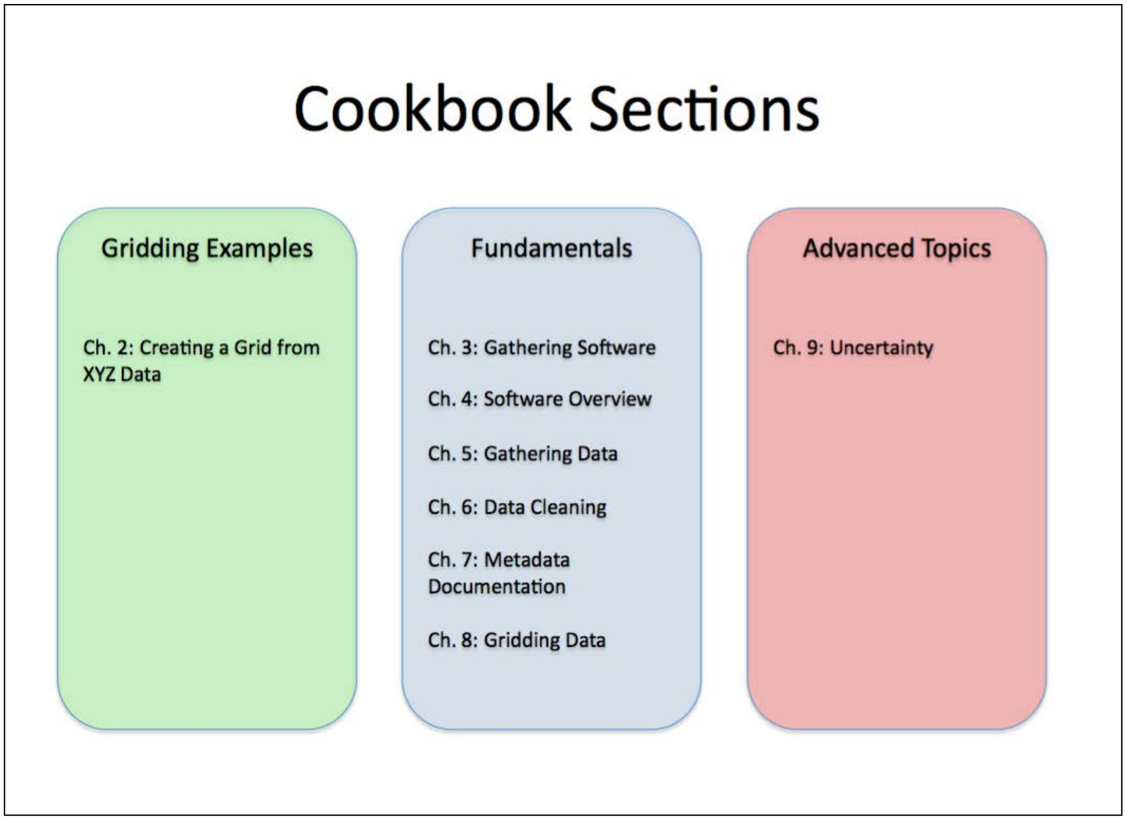
GEBCO, the General Bathymetric Chart of the Oceans, is a joint project of the International Hydrographic Organization (IHO) and the Intergovernmental Oceanographic Commission (IOC) bringing together an international group of experts whose aim is to provide the most authoritative publicly available bathymetry of the world's oceans, for scientific and educational use. GEBCO welcomes and encourages new contributions of data and expertise from all over the world.

One of GEBCO's products is a global gridded bathymetry data set. This grid contains depth (and, on land, elevation) estimates on a 30 arc-second mesh, which near the equator is about a one-kilometer spacing. Over the oceans, seafloor depths are from ship soundings, and where there are no soundings depths are interpolated, using estimates from satellite altimetry as a guide. The global bathymetry grid is periodically updated by incorporating additional bathymetric survey data, new compilations, and improved data. With this document, the "GEBCO Cook Book," we hope to enable more people to contribute data and gridded compilations to GEBCO.

The Cook Book contains chapters that span basic to advanced topics. These have been written by expert GEBCO contributors from more than a dozen international research organizations, universities, governments, and companies. The Cook Book is a "living document" that will be continually updated and expanded as new contributions are received, and as the GEBCO community's sense of best practices evolves. The Cook Book is available for free download from the GEBCO website (http://www.gebco.net).

Approach

The Cook Book is composed of three main sections- "Gridding Examples," "Fundamentals," and "Advanced Topics."



We start with gridding examples so that beginning users desiring to straightaway produce a grid from xyz data can do so by delving into the first section. We make available an example xyz data set through Internet download and also show an alternate method to obtain it through the National Geophysical Data Center (NGDC) website. A brief overview of gridding software packages, some of which are freely available for download from the Internet, is provided. Illustrated step-by-step instructions that include command lines to process and grid xyz data using the various software packages are listed. The command lines can be cut-and-paste onto the user's computer and, along with the example xyz data, the user can reproduce the results shown in the Cook Book and then apply the same technique to their own data. The results are displayed in maps and plots that can be reproduced with the given command lines as well. This step-by-step approach also allows us to encourage users, once they are familiar with running the example command lines and producing grids and maps, to try changing the command options so that they can see for themselves how the changes affect the final result.

The second section, "Fundamentals," gives a more in-depth look at topics related to preparing, processing, and gridding xyz data. An overview of various gridding software packages, including ArcGIS, CARIS HIPS, Generic Mapping Tools (GMT), r2v, Surfer, and Global Mapper, is included. Also covered are the topics of data cleaning, assessing data quality, validation, metadata documentation, and discussion on the various gridding techniques.

The third section, "Advanced Topics," contains discussions that more experienced users may find useful. Uncertainties are covered in detail- its sources, and methods of measurement including Monte Carlo and Network techniques, windowed regression, error growth models, and more. Methods of sharing uncertainty results in the form of ASCII, netCDF, and "BAG" files is included.

Lastly, three Annexes are provided. Annex on Additional Resources enables the user to find more information and includes a list of websites from institutions and organizations that make multibeam data available for public Internet download. An Annex containing a Glossary as well as an Annex with Acronyms and Abbreviations are also provided.

We note that the IHO, IOC, GEBCO, and the contributing authors and institutions do not endorse or promote any particular software package or software provider. The information in this document is provided to benefit the user in good faith but the accuracy and completeness cannot be guaranteed.

GRIDDING EXAMPLES

Chapter 2.0 Creating a Grid from XYZ Data

This chapter provides step-by-step instructions that will show a user how to process and grid xyz data. An example data set is made available so the user can follow along on their own computer.

2.1 Gridding XYZ Data with Generic Mapping Tools (GMT)

Contributed by K. M. Marks, NOAA Laboratory for Satellite Altimetry, USA
Updated by Karolina Chorzewska, University of New Hampshire, USA

2.1.1 Downloading Example Data Set

In this section we demonstrate how to create a grid from xyz bathymetry data. The data set used in this demonstration is available for download from
ftp://ibis.grdl.noaa.gov/pub/karen/outgoing/ngdc.topo.xyz

These data may also be downloaded directly from the NOAA National Geophysical Data Center (NGDC) website as follows. First, access the GEODAS (GEOphysical DAta System) webpage at http://www.ngdc.noaa.gov/mgg/gdas/, which is shown in Fig. 2.1.

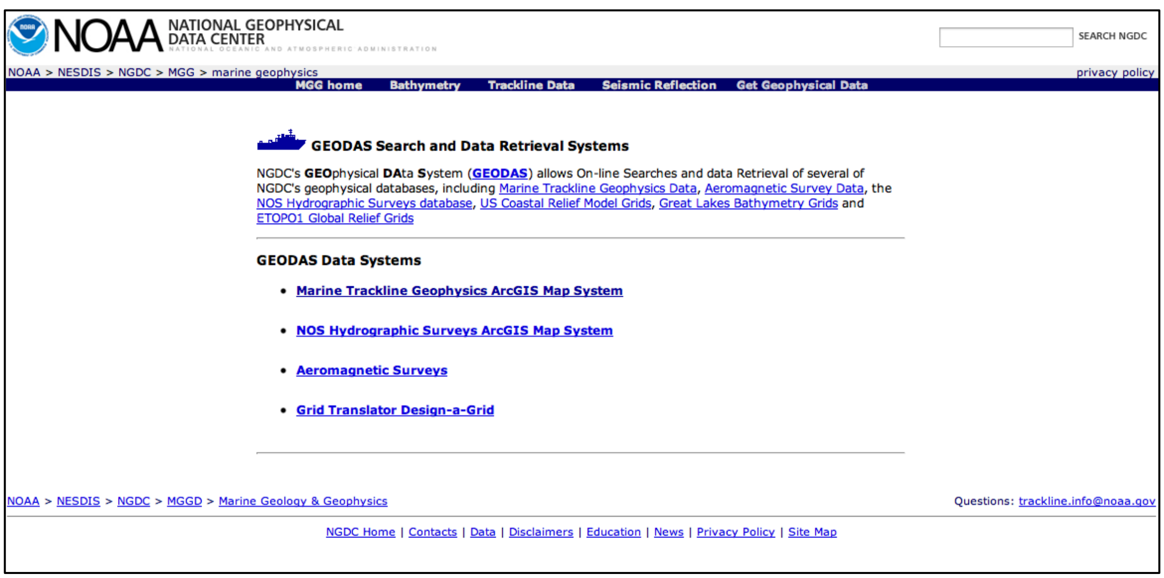


Figure 2.1 NOAA National Geophysical Data Center (NGDC) GEODAS website

Select "Marine Trackline Geophysics ArcGIS Map System" to reach the Geophysical Survey Data web page (see Fig. 2.2 below).

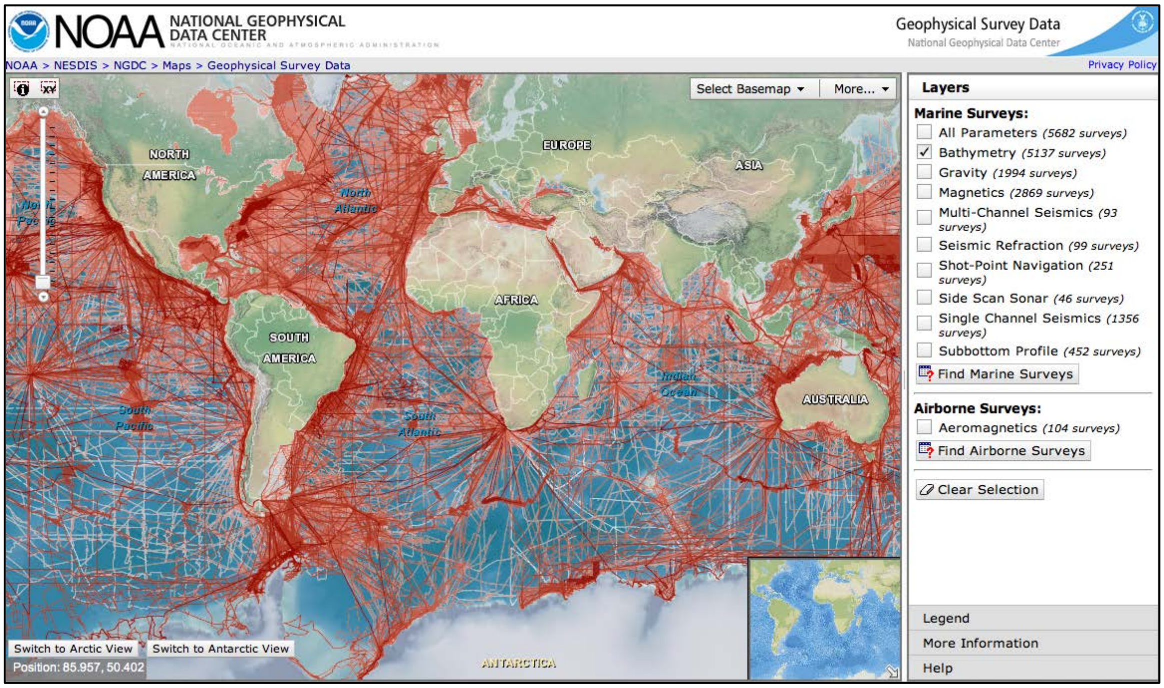


Figure 2.2 Geophysical Survey Data web page.

Check “Bathymetry” in the “Layers” window on the right side of the screen (Fig. 2.2). Click the XY icon (“Identify with Coordinates”) in the top-left corner of the map (also Fig. 2.2). It will open a “Specify an Area of Interest” window (see Fig 2.3 below). Our data set ranges from 15 to 20 N latitude, and from 150 to 154 E longitude. The user should enter these geographic coordinates into the corresponding boxes, and then press “OK.”

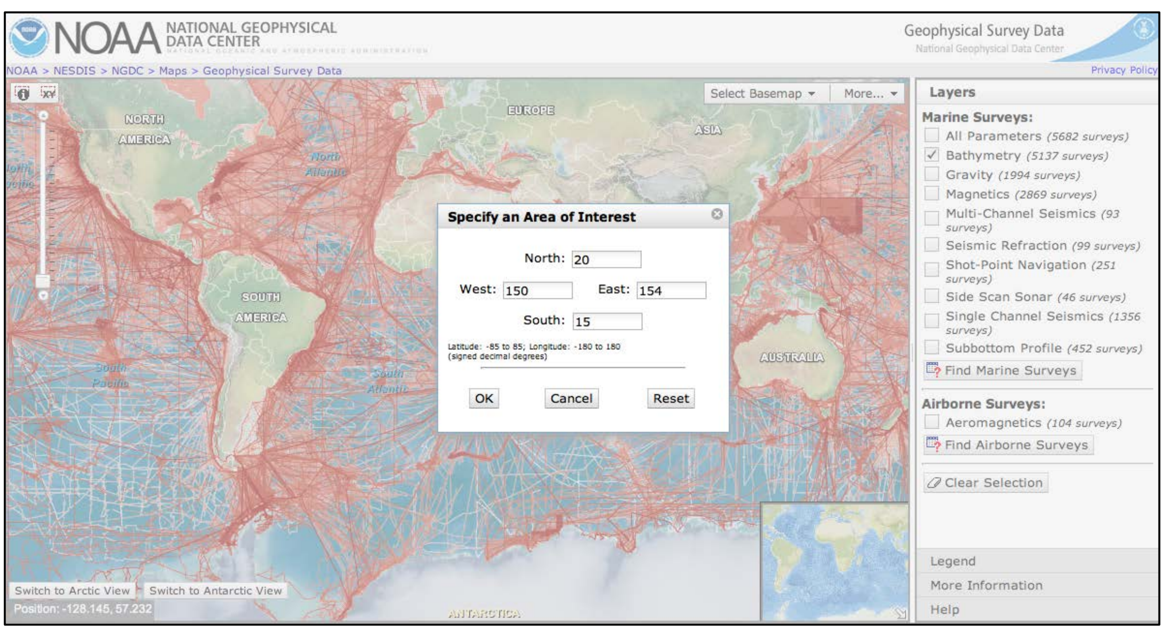


Figure 2.3 Geographic boundaries selection window.

The specified area's boundary is now shown on the map and available surveys are listed in the "Identified Features" window (see Fig. 2.4 below).

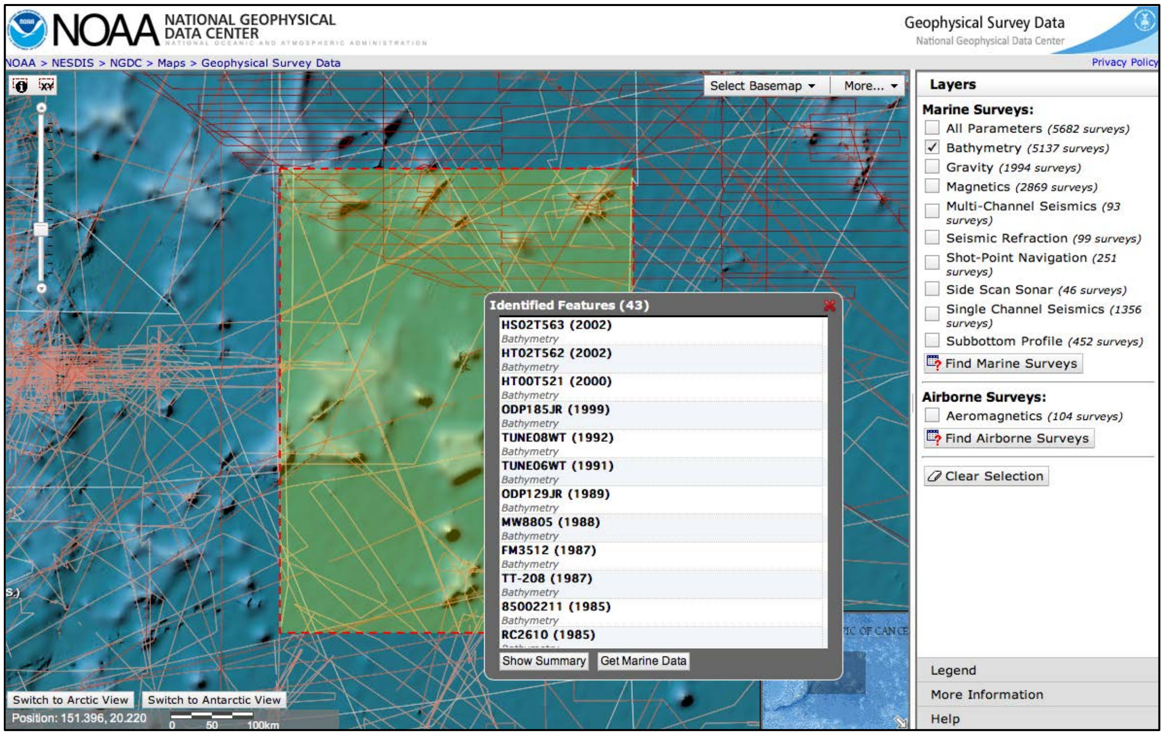
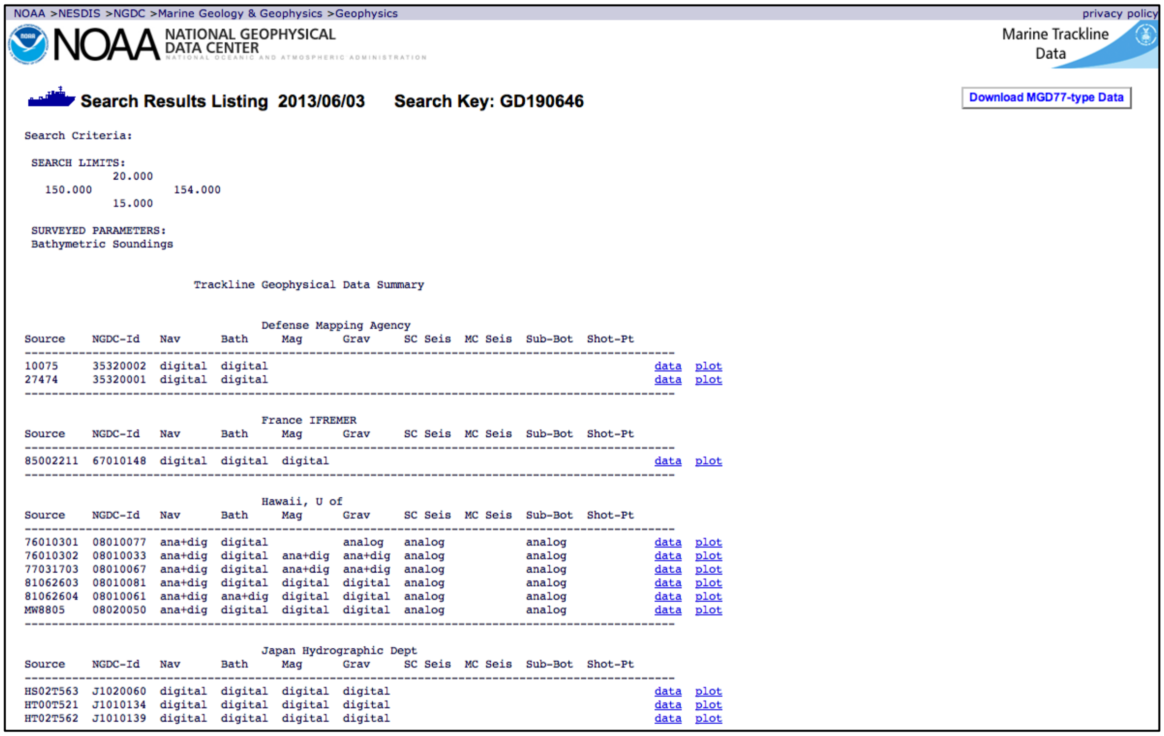


Figure 2.4 Search results image.

Select "Get Marine Data." The results of the search are displayed (see Fig. 2.5 below). Select "Download MGD77-type Data."



NOAA >NESDIS >NGDC >Marine Geology & Geophysics >Geophysics privacy policy

NOAA NATIONAL GEOPHYSICAL DATA CENTER

Marine Trackline Data

Search Results Listing 2013/06/03 Search Key: GD190646

Download MGD77-type Data

Search Criteria:

SEARCH LIMITS:
20.000
150.000 154.000
15.000

SURVEYED PARAMETERS:
Bathymetric Soundings

Trackline Geophysical Data Summary

Defense Mapping Agency

Source	NGDC-Id	Nav	Bath	Mag	Grav	SC Seis	MC Seis	Sub-Bot	Shot-Pt		
10075	35320002	digital	digital							data	plot
27474	35320001	digital	digital							data	plot

France IFREMER

Source	NGDC-Id	Nav	Bath	Mag	Grav	SC Seis	MC Seis	Sub-Bot	Shot-Pt		
85002211	67010148	digital	digital	digital						data	plot

Hawaii, U of

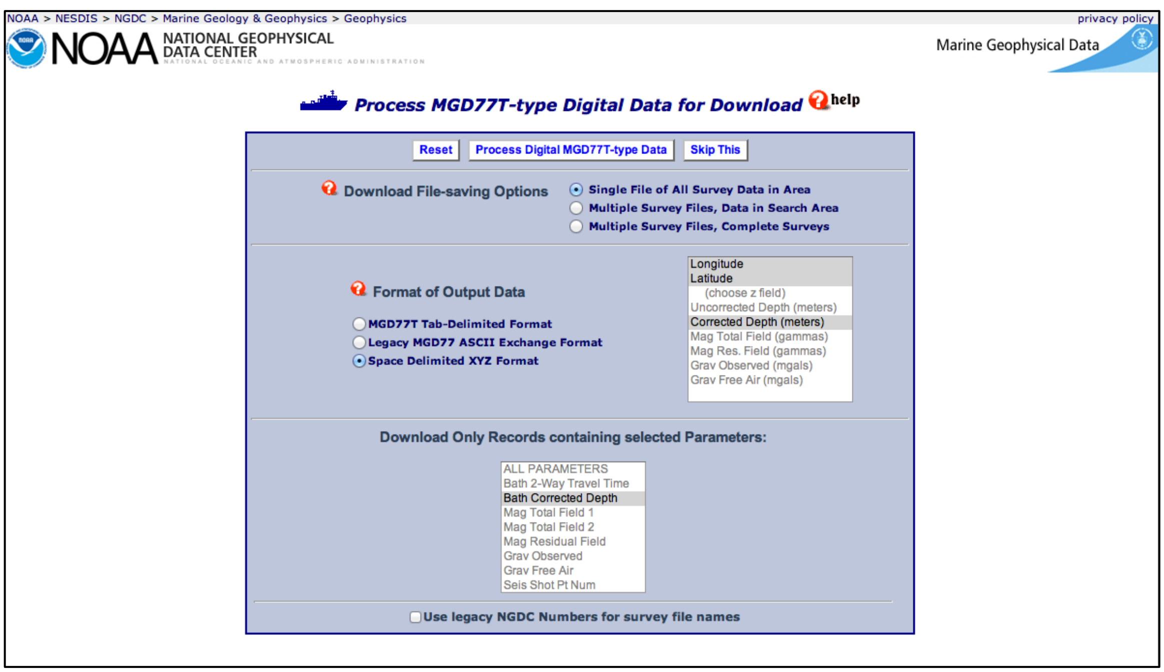
Source	NGDC-Id	Nav	Bath	Mag	Grav	SC Seis	MC Seis	Sub-Bot	Shot-Pt		
76010301	08010077	ana+dig	digital		analog	analog		analog		data	plot
76010302	08010033	ana+dig	digital	ana+dig	ana+dig	analog		analog		data	plot
77031703	08010067	ana+dig	digital	ana+dig	ana+dig	analog		analog		data	plot
81062603	08010081	ana+dig	digital	digital	digital	analog		analog		data	plot
81062604	08010061	ana+dig	ana+dig	digital	digital	analog		analog		data	plot
MW8805	08020050	ana+dig	digital	digital	digital	analog		analog		data	plot

Japan Hydrographic Dept

Source	NGDC-Id	Nav	Bath	Mag	Grav	SC Seis	MC Seis	Sub-Bot	Shot-Pt		
HS02T563	J1020060	digital	digital	digital	digital					data	plot
HT00T521	J1010134	digital	digital	digital	digital					data	plot
HT02T562	J1010139	digital	digital	digital	digital					data	plot

Figure 2.5 Portion of Search Results web page.

The desired format of the downloaded data can be designated next (see Fig. 2.6 below). For our data set, select "Single File of All Survey Data in Area," "Space Delimited XYZ Format," and select Longitude, Latitude, and Corrected Depths (meters) in the box. Then press the "Process Digital MGD77T-type Data" button.



NOAA > NESDIS > NGDC > Marine Geology & Geophysics > Geophysics privacy policy

NOAA NATIONAL GEOPHYSICAL DATA CENTER

Marine Geophysical Data

Process MGD77T-type Digital Data for Download help

Reset | Process Digital MGD77T-type Data | Skip This

Download File-saving Options
- Single File of All Survey Data in Area
- Multiple Survey Files, Data in Search Area
- Multiple Survey Files, Complete Surveys

Format of Output Data
- MGD77T Tab-Delimited Format
- Legacy MGD77 ASCII Exchange Format
- Space Delimited XYZ Format

Longitude
Latitude
(choose z field)
Uncorrected Depth (meters)
Corrected Depth (meters)
Mag Total Field (gammas)
Mag Res. Field (gammas)
Grav Observed (mgals)
Grav Free Air (mgals)

Download Only Records containing selected Parameters:

ALL PARAMETERS
Bath 2-Way Travel Time
Bath Corrected Depth
Mag Total Field 1
Mag Total Field 2
Mag Residual Field
Grav Observed
Grav Free Air
Seis Shot Pt Num

Use legacy NGDC Numbers for survey file names

Figure 2.6 Download Format web page.

A web page notifying the user that the data have been processed and are available for retrieval comes up next (see Fig. 2.7 below). Press the "Compress and Retrieve Your Data" button. The GEODAS system has assigned the arbitrary name "GD190646" to the data set.

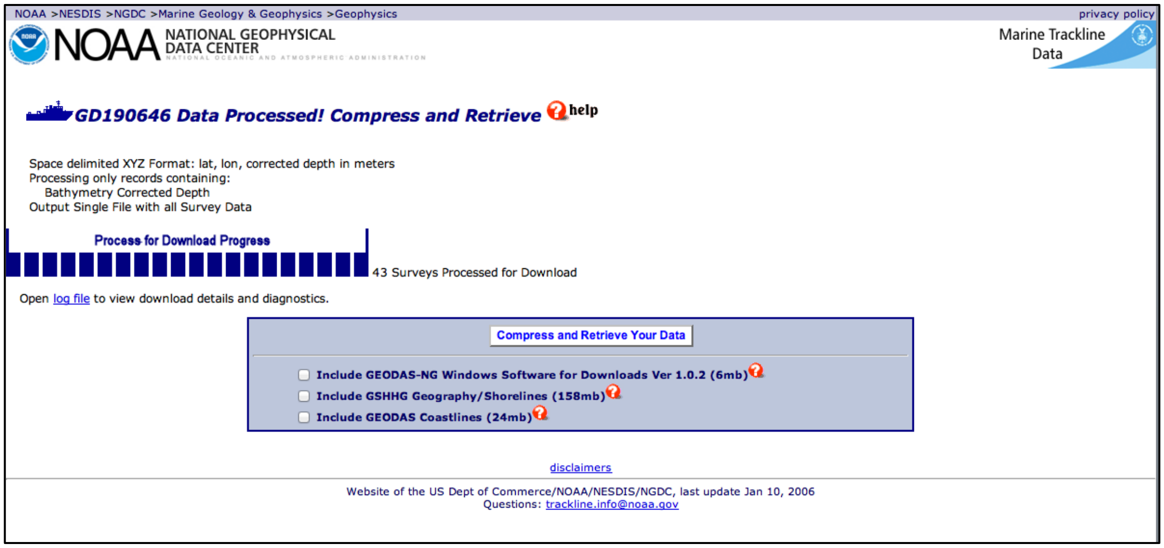


Figure 2.7 Compress and Retrieve data web page.

The next web page provides information about retrieving the data (see Fig. 2.8 below). Push the "Retrieve" button to download the data set. If the download file size is particularly large, select "Retrieve Compressed File" next to the retrieve button to reduce the size of the download file. The data file is then downloaded onto the user's computer.

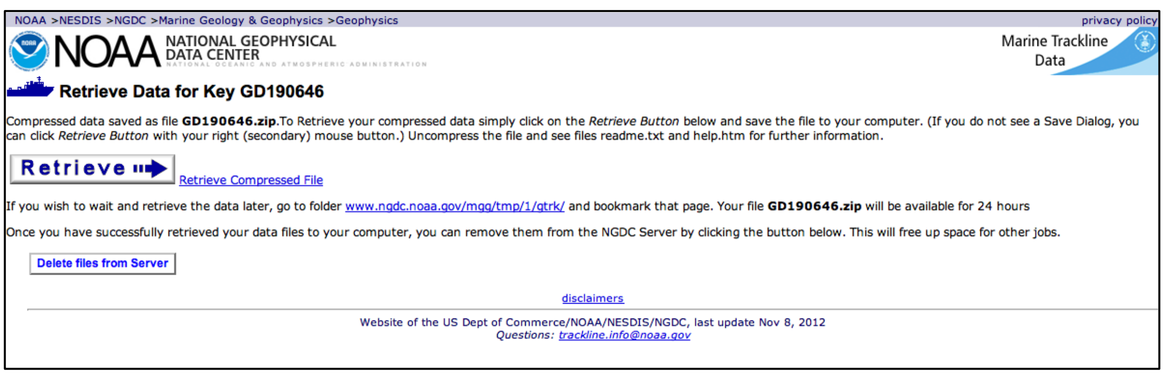


Figure 2.8 Retrieve download.

The data are downloaded (or uncompressed) as a directory (e.g., geodas_ GD190646) on the user’s computer. Within the directory are documents, readme information, and the data file (GD190646.xyz). We chose to rename the xyz data set "ngdc.topo.xyz."

2.1.2 *Generic Mapping Tools (GMT)*

GMT (Generic Mapping Tools) (Wessel and Smith, 1998) is a collection of open source mathematical and mapping routines for use on gridded data sets, data series, and arbitrarily located data. The GMT package is available for download from the University of Hawaii website (http://gmt.soest.hawaii.edu/) (see Fig. 2.9). We utilized GMT routines for our gridding and mapping examples shown in this section. Actual GMT command lines are provided in the discussions that follow.

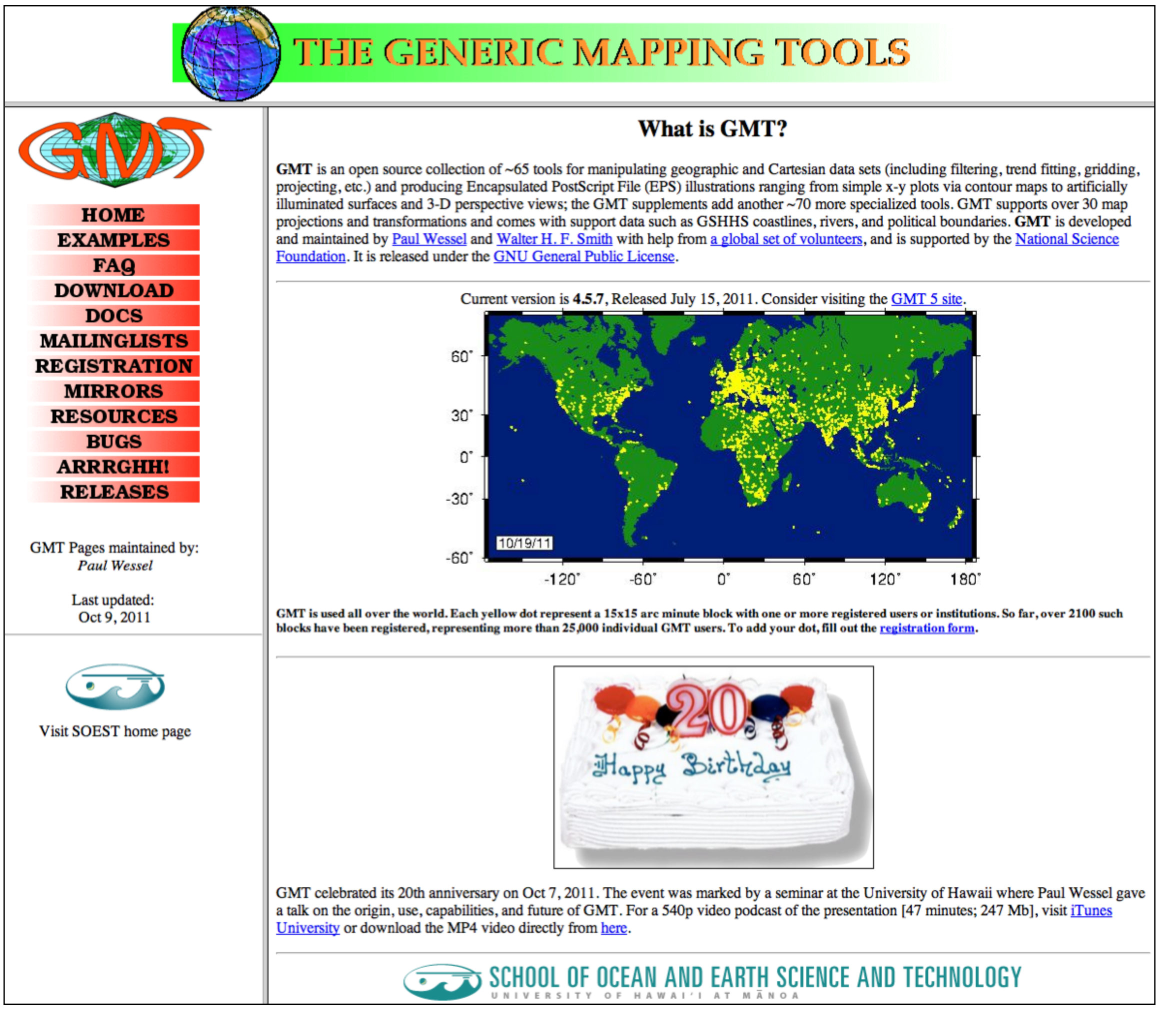


Figure 2.9 University of Hawaii website for GMT.

2.1.3 *Plotting Data with GMT*

It is desirable to plot the xyz data on a map to see the locations of the points. The xyz data may be plotted on a map using Generic Mapping Tools (GMT).

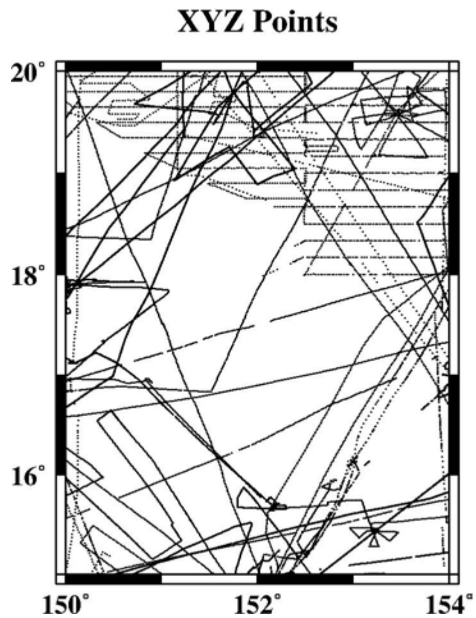


Figure 2.10 Map of XYZ point locations.

Figure 2.10 (above) was produced with the following GMT commands:

```
psbasemap -R150/154/15/20 –Jm0.5 -Ba2f1:."XYZ Points":WeSn -K > Fig.2.10.ps
psxy ngdc.topo.xyz -R -Jm -Sp -O >> Fig.2.10.ps
```

It is useful to note the distribution of the data points. In the center of the map there are few ship tracks, leaving large areas with no bathymetry measurements. In the northeastern corner, there is systematic survey coverage.

2.1.4   Gridding XYZ Data with GMT

It is possible to create a grid from the randomly located xyz data points. GMT routine "surface" is an adjustable tension continuous curvature gridding algorithm that can be used to grid the xyz data. We input the xyz data into "surface," setting the tension factor to 0.25 (surface tension set to "0" gives the minimum curvature solution, and set to "1" gives a harmonic surface where maxima and minima are only possible at control points). Figure 2.11 (below) shows the resulting grid.

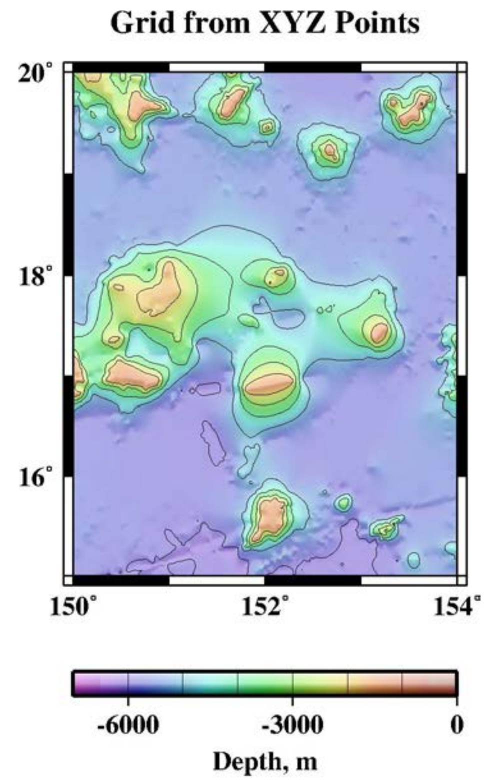


Figure 2.11 Grid of XYZ data points, using GMT "surface" with tension set to 0.25.

The GMT commands used to produce the gridding solution and plot it in Fig. 2.11 follow.

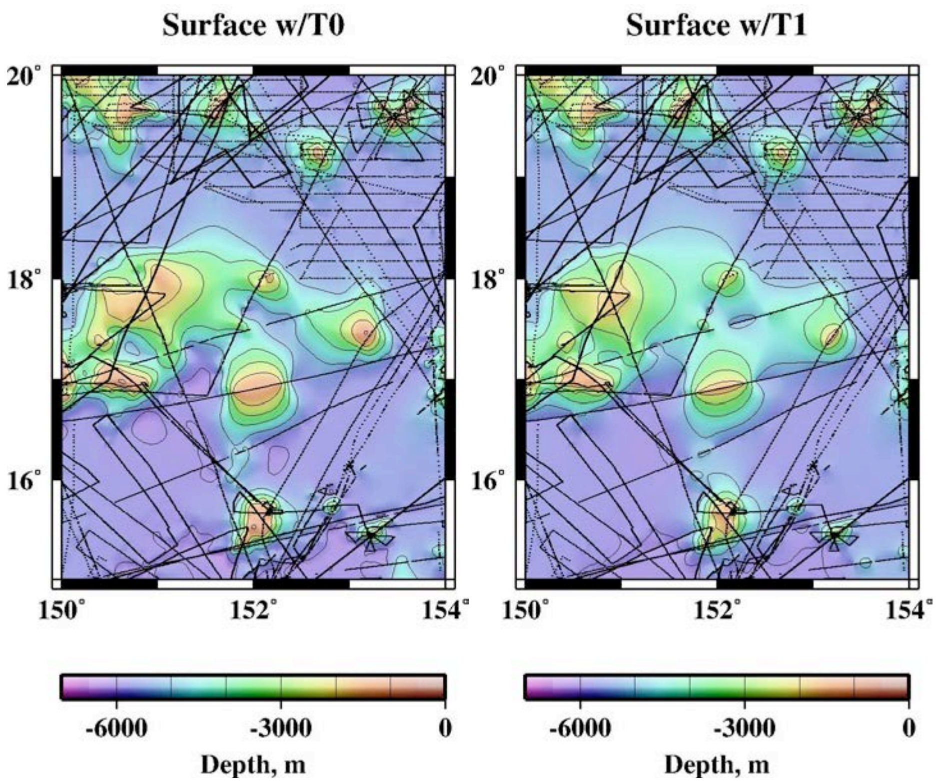
```
surface ngdc.topo.xyz -R150/154/15/20 -I1m -T0.25 -Gngdc.topo_grd
grdgradient ngdc.topo_grd -A0 -Ne0.2 -Ggrad_grd
grdimage ngdc.topo_grd -Igrad_grd -Ctopo.cpt -Jm0.5 -K>Fig.2.11.ps
pscoast -G175 -R150/154/15/20 -Jm0.5 -W2 -Df -Ba2f1:."Grid from XYZ Points":WeSn
        -K -O>>Fig.2.11.ps
grdcontour ngdc.topo_grd -R -Jm -C1000 -W1 -K -O>>Fig.2.11.ps
psscale -D1/-.5/2/.125h -Ctopo.cpt -Ba3000g1000:"Depth, m": -I -N300 -O >>Fig.2.11.ps
```

The color palette table (topo.cpt) is:

```
-7000   300        .350000 .85   -6500   275.625      .350000 .85
-6500   275.625    .300000 .90   -6000   250.3125     .300000 .90
-6000   250.3125   .300000 .90   -5500   225          .300000 .90
-5500   225        .300000 .90   -5000   199.6875     .300000 .90
-5000   199.6875   .300000 .90   -4500   175.3125     .300000 .95
-4500   175.3125   .300000 .95   -4000   150          .350000 .95
-4000   150        .350000 .95   -3500   125.625      .350000 .95
-3500   125.625    .350000 .95   -3000   99.375       .350000 .95
-3000   99.375     .350000 .95   -2500   75           .350000 .95
-2500   75         .350000 .95   -2000   50.625       .350000 .95
-2000   50.625     .350000 .95   -1500   25.3125      .350000 .95
-1500   25.3125    .350000 .95   -500    10           .250000 .85
-500    10         .200000 .85   0       10           .150000 .80
B       300        .350000 1
F       0          0.00000 1.0
```

The color palette file above specifies the depth ranges (in meters) over which colors are assigned by the HSV (Hue, Saturation, Value) color system. The color palette file is entered into GMT routine “grdimage” above via the “-C” command line argument. Online GMT documentation (http://gmt.soest.hawaii.edu) provides details about color palette files and various color systems.

There are other algorithms that can be used to grid the xyz data points, and each will produce a different solution; the differences are most evident in areas where there are large gaps between measurements. Changing the tension parameter in GMT "surface" will also produce a different solution. "Nearneighbor" uses a nearest neighbor algorithm to assign an average value within a radius centered on a node. "Triangulate" performs optimal Delaunay triangulation. We show the results of GMT "surface," "nearneighbor," and "triangulate" gridding algorithms below.



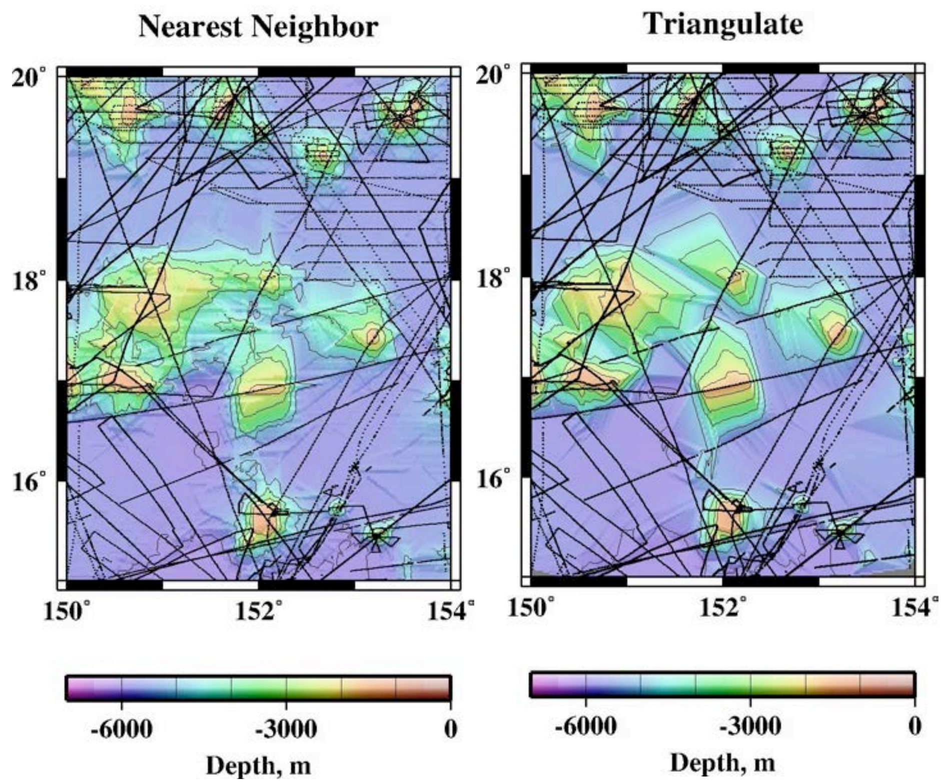


Figure 2.12 Results of gridding XYZ data (black dots) with GMT gridding routines "surface" with tension set to 0 (top left) and set to 1 (top right), "nearneighbor" (bottom left), and "triangulate" (bottom right).

The command lines used for the corresponding GMT gridding routines follow. The commands to plot the figures are the same as listed for Fig. 2.11, and the color palette file is the same.

```
surface ngdc.topo.xyz -R150/154/15/20 -I1m -T0 -Gngdc.topo_grd
surface ngdc.topo.xyz -R150/154/15/20 -I1m -T1 -Gngdc.topo_grd
nearneighbor ngdc.topo.xyz -R150/154/15/20 -N4/1 -S100k -I1m -Gngdc.topo_grd
triangulate ngdc.topo.xyz -R150/154/15/20 -I1m -Gngdc.topo_grd > junk
```

Each gridding algorithm produces a very different result because each one uses a different method of interpolating depth estimates in the gaps between data points. Where the gaps are large, areas of shallower seafloor (orange-yellow colored anomalies) appear to follow the ship tracks. This is because ships collect depth measurements along one-dimensional tracks without detecting seafloor away from the track. Seafloor can be mapped in two dimensions when the survey coverage is adequate (for example, in the northeast corner of the study area).

Bathymetry models that estimate depths from a combination of ship measurements and depths derived from satellite gravity can also map the seafloor in two dimensions. Fig. 2.13 below shows the Smith and Sandwell (1997) bathymetry model (version 13.1) over our study area.

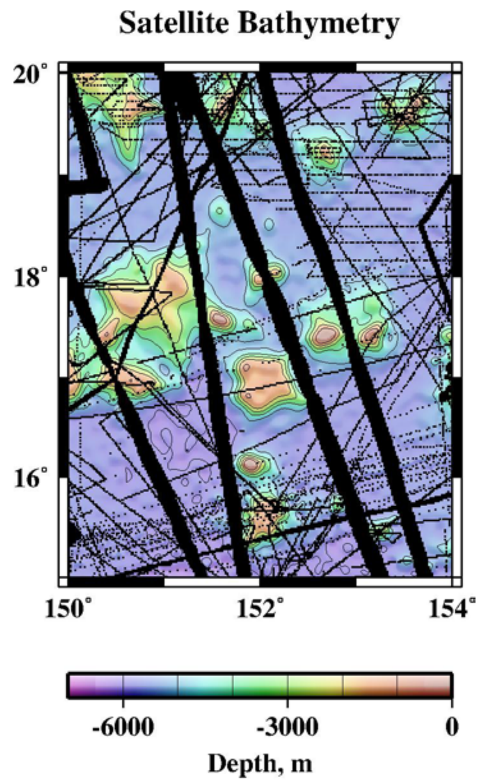


Figure 2.13 Grid of estimated depths from altimetric bathymetry model, black dots are ship controls used in model.

It is important to note that some bathymetry anomalies lie in gaps between ship controls (black dots), and are also mapped away from the ship tracks. Because bathymetry models incorporate depths estimated from satellite gravity and the satellite tracks have ~4 km uniform spacing over the oceans, bathymetry anomalies are accurately mapped in two dimensions. Grids created only from xyz data containing large gaps may not accurately map the seafloor located between the ship tracks.

In our experience, GMT "surface" with a tension set to 0.25 produces a nice-looking bathymetry grid.

2.1.5 What does a Bad Track in a Grid look like?

Gridding algorithms simply ingest the xyz data supplied, and produce a grid. If the xyz data contain errors, they may be visible in the resulting grid. A common problem is a "bad track," where the ship data along a track are poor, which can result from a variety of reasons. To demonstrate what a "bad track" in a grid looks like, we added 2000 m to the depths of track 01010060 (NGDC ID number) and combined it with the example xyz data set and gridded it using routine GMT "surface" (see Fig. 2.14).

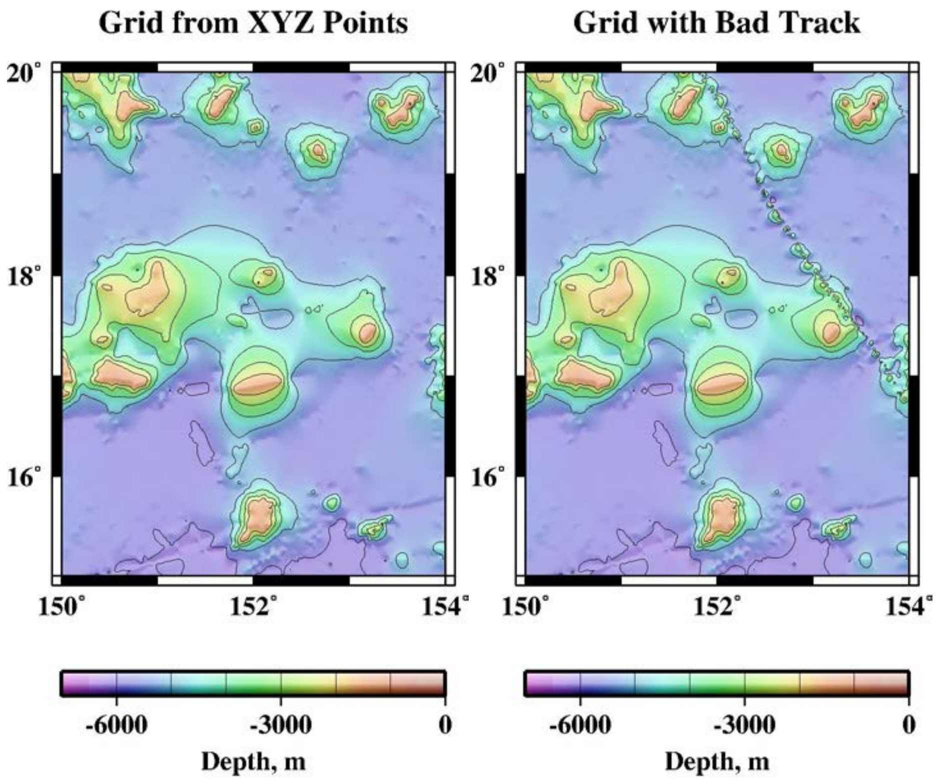


Figure 2.14 Grid of XYZ data points (left), and grid of same XYZ data points with "bad track" included (right).

In the right panel of Fig. 2.14, the bad track stands out as a line of local shallow seafloor anomalies. Contours adjacent to the line are contorted and jagged. Sometimes a bad track or other data with errors can be easy to find as in this demonstration, but they can also be subtle. It is important to use data that are as clean and free of errors as possible.

2.2 Gridding XYZ Data with ArcMap

Contributed by Anastasia Abramova, Geological Institute Russian Academy of Sciences, Russia
Updated by Karolina Chorzewska, University of New Hampshire, USA

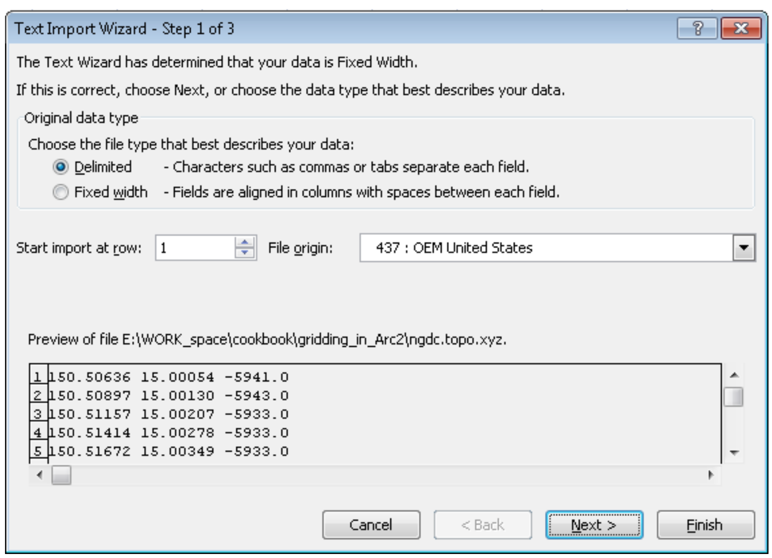
Here we present step by step example on how to open .xyz space delimited column data in ArcMap and to interpolate data points into a grid of appropriate resolution. As an example ngdc.topo.xyz file (See Section 2.1.1) in geographic coordinate system is used. We would like to create a grid in UTM projection with appropriate cell resolution using Inverse Distance Weighting Interpolation algorithm. The procedure of reprojecting the data, assessing the data density to choose suitable grid size is described. Tools available in ArcMap in order to select optimal interpolation parameters and to compare resultant surface with original data points are discussed. In order to be able to carry out above noted you will need to have Spatial Analyst and Geostatistical Analyst extensions.

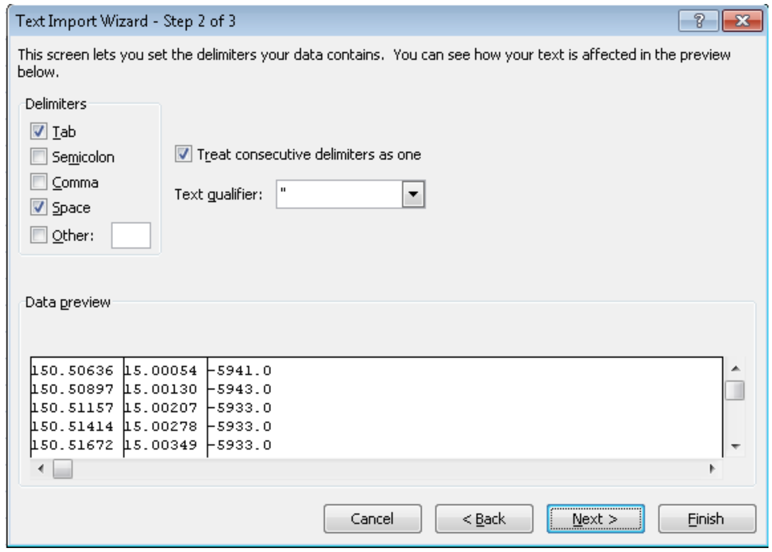
2.2.1 Importing XYZ file in ArcMap

In order to open ngdc.topo.xyz file in Arc, you need to modify it: text file needs to have header, data should be comma separated and file name should not have spaces in it.

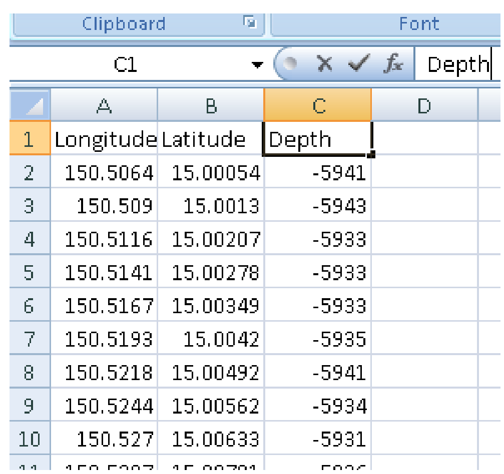
In order to modify ngdc.topo.xyz text file, import it into MS Excel:

- File>Open ngdc.topo.xyz
- Select Delimited, press Next

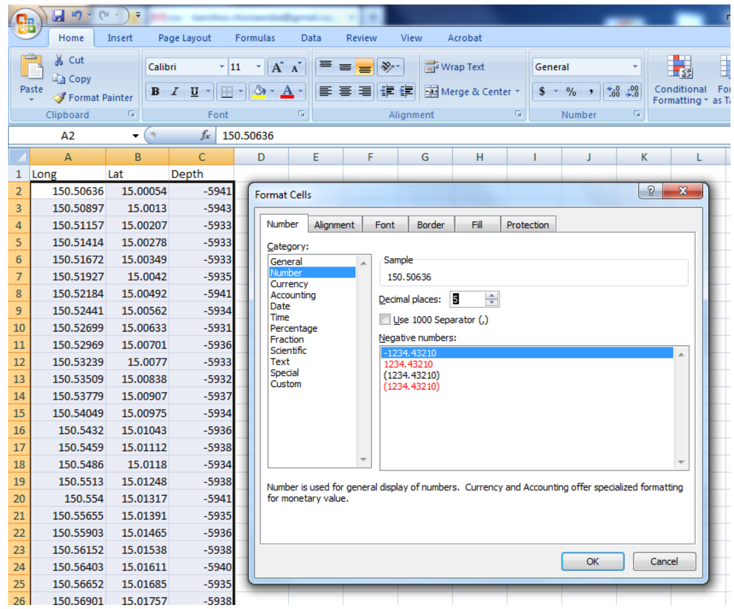




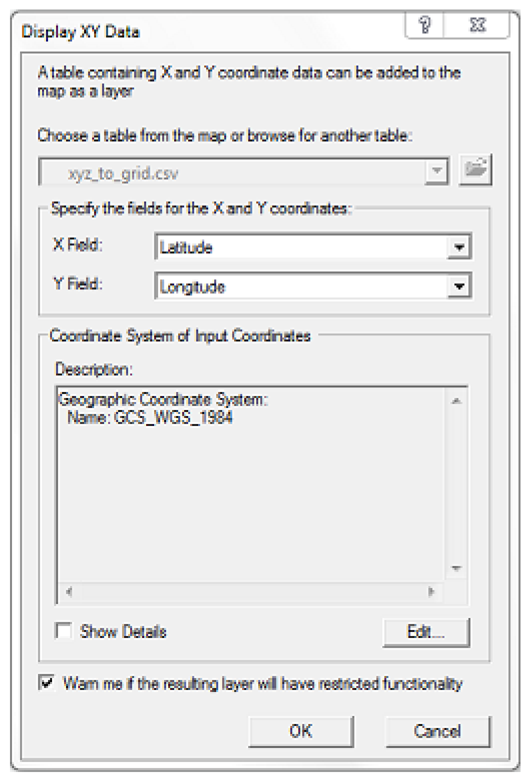
- Insert header row, type header name for each column (no spaces!)



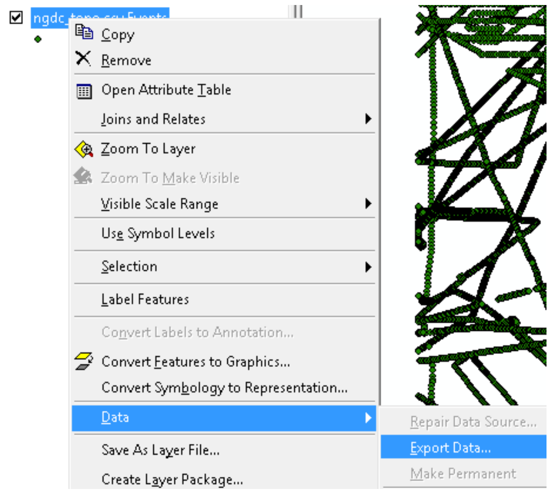
- Make sure that the cells that contain numbers are formatted as Numbers in MS Excel
- File> Save as ngdc_topo.dbf (*.dbf) (or *.xls will work too)
- Close file In MS Excel! (Arc doesn't want to open it if it is opened somewhere else)



- Open ArcMap: File> New > Blank Document
- Go to Tools> Add XY data and add ngdc_topo.dbf file
- Select the appropriate fields: X as Latitude and Y as Longitude. In order to define the input data projection: press Edit > Select> Geographic> World > WGS 1984. Press Ok



- Now we need to save file as ArcMap shape file, right click on the Layer> Data> Export Data> Save as ngdc_topo.shp file (this is the file we will be working with further on)

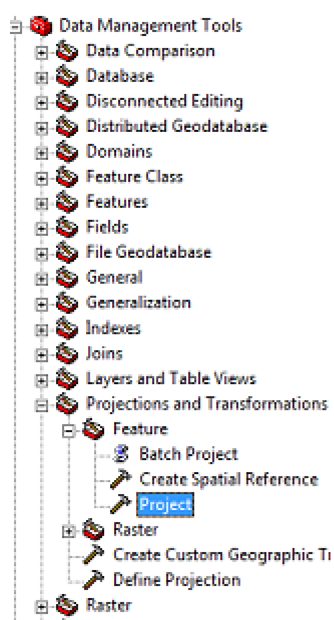


2.2.2 *Reprojecting the data*

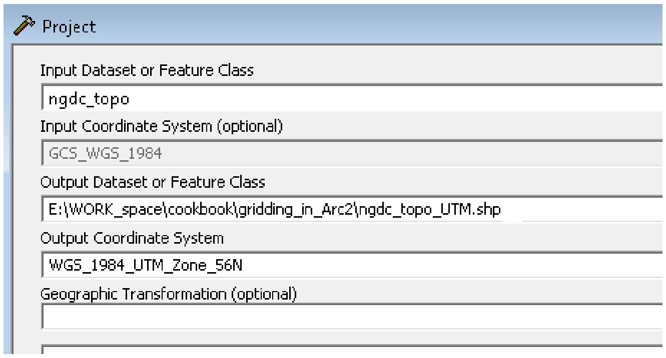
- We would like to create a grid in UTM projection with units in meters, instead of degrees. For that we need to reproject our data.

- Click on Show/Hide ArcToolbox:

- In opened ArcToolbox choose Data Management Tools> Projections and Transformations> Feature> Project



- Define the Input and Output files (ngdc_tpo_UTM.shp) and Output Coordinate System. In our case it is Projected Coordinate System>UTM>WGS1984>UTM_Zone_56N). Press Ok



- You might want to change the data frame projection view from Geographic to UTM_Zone_56N now (projection in the data frame view is defined by projection of the first opened file). Right click on Layers> Properties> Coordinate Systems. Select from Predefined list Projected> UTM> WGS 1984> UTM Zone 56N. Press Ok. Now data frame view has the same projection as reprojected file.

- Finally our added data looks like this. As you can notice, the data point density varies over the area

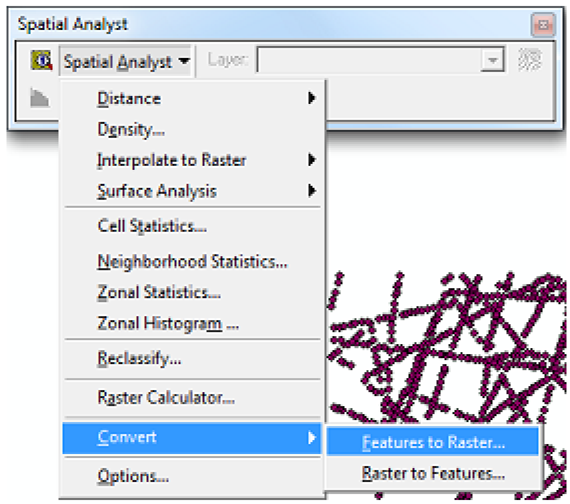
2.2.3 Defining cell resolution for the grid

- The choice of grid resolution depends on your purpose. Meanwhile, data distribution, density, complexity of modeled terrain and scale of the final map should be taken into account as well.

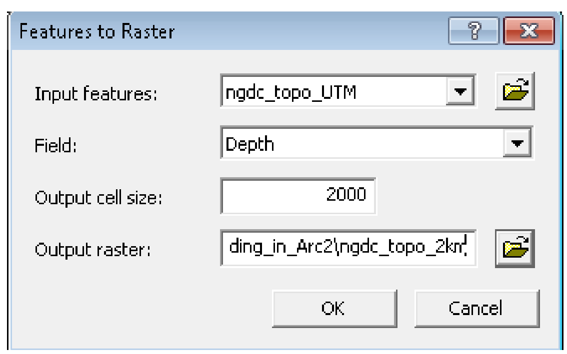
- We would like to create grid with 2 km cell resolution. Here we show how to identify the percentage of cells which values will be defined by interpolation and how to estimate the data density within each cell for the chosen grid size.

- Make sure that Spatial Analyst Extension is activated: go to Tools> Extensions> check Spatial Analyst. Also, add Spatial Analyst toolbar: go to View> Toolbars> check Spatial Analyst.

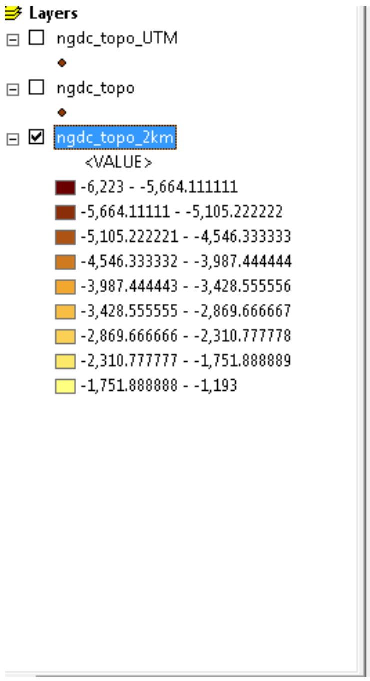
- From Spatial Analyst toolbar select Convert> Features to Raster

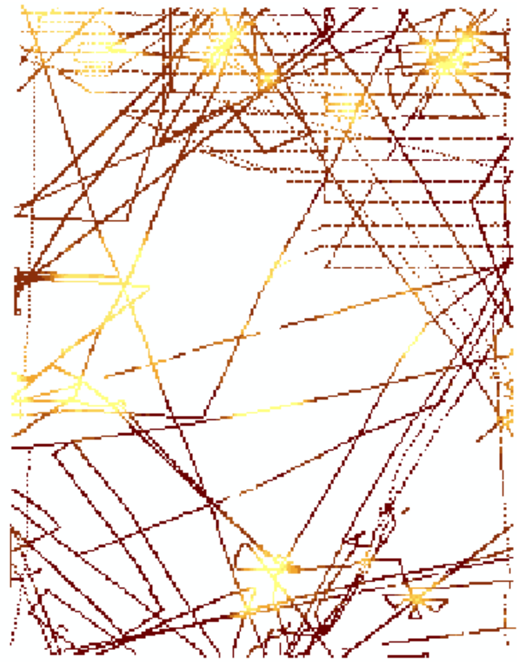


- Select appropriate field and cell resolution:

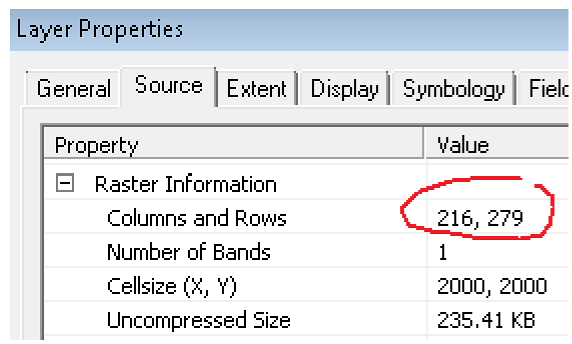


- As you can see, in the created raster (ngdc_topo_2km), only cells with data points in it have non zero values:

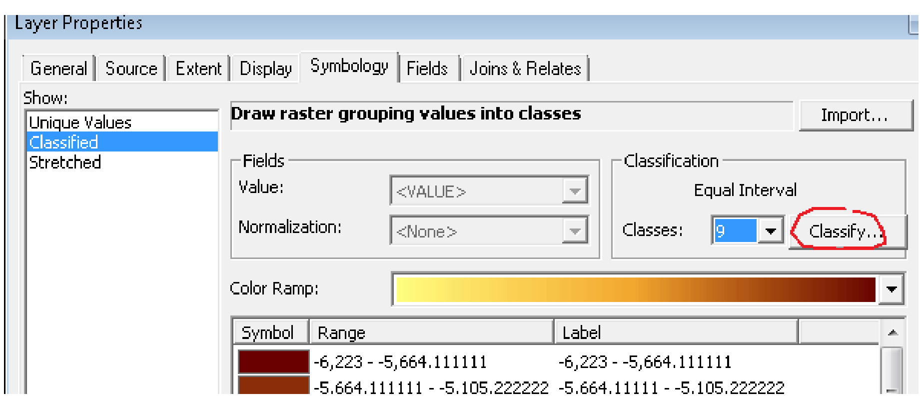




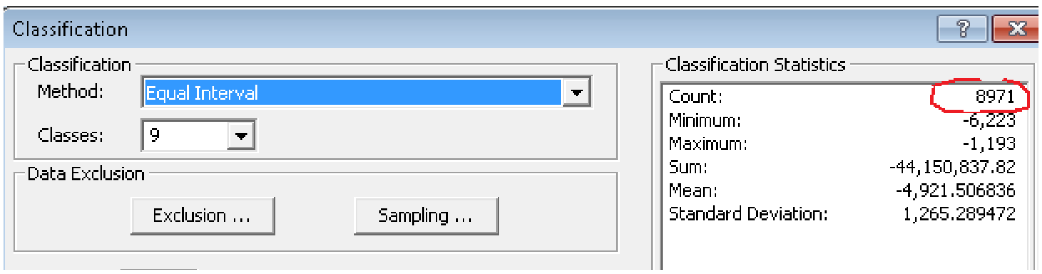
- In order to find out what is the percentage of cells with "no data values", go to Layer> Properties> Source. Here you can look up the total number of cells in the grid created by multiplying total number of rows by number of columns (216*279=60264)



- In order to find out how many cells have non zero values, go to Symbology tab in Layer Properties window and click on Classified. Select Classify tab:

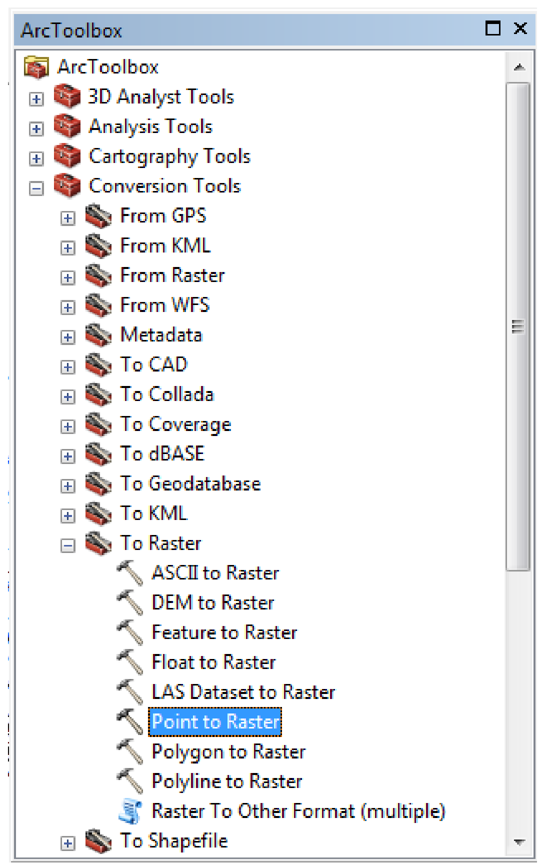


- The classification table will appear. In Classification Statistics Window, Count will tell the amount of non zero cells (8971). This is approximately 14 % of total amount of cells. Therefore, the values for 86% of cells will have to be estimated by interpolation at chosen resolution of 2 km:

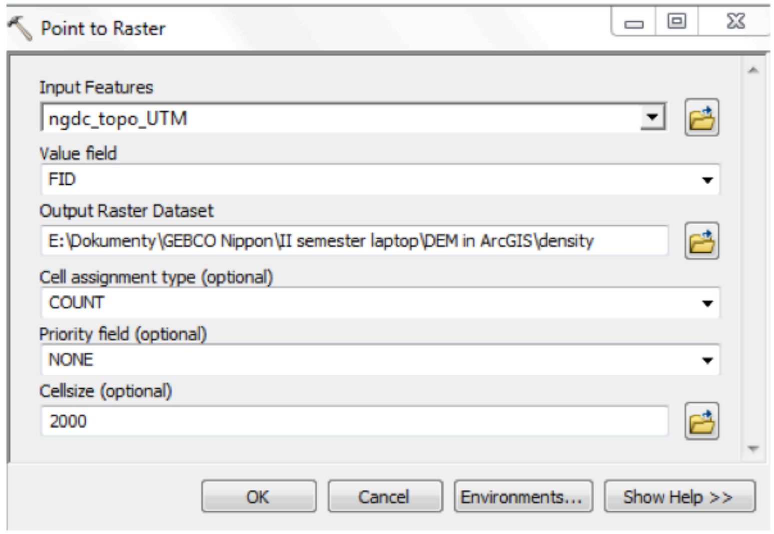


- The density of data points per cell for the chosen grid size should be checked. Number of data points per one cell depends on your purpose.

- Choose Point to Raster tool from ArcToolbox (ArcToolbox → Conversion Tools → To Raster → Point to Raster).

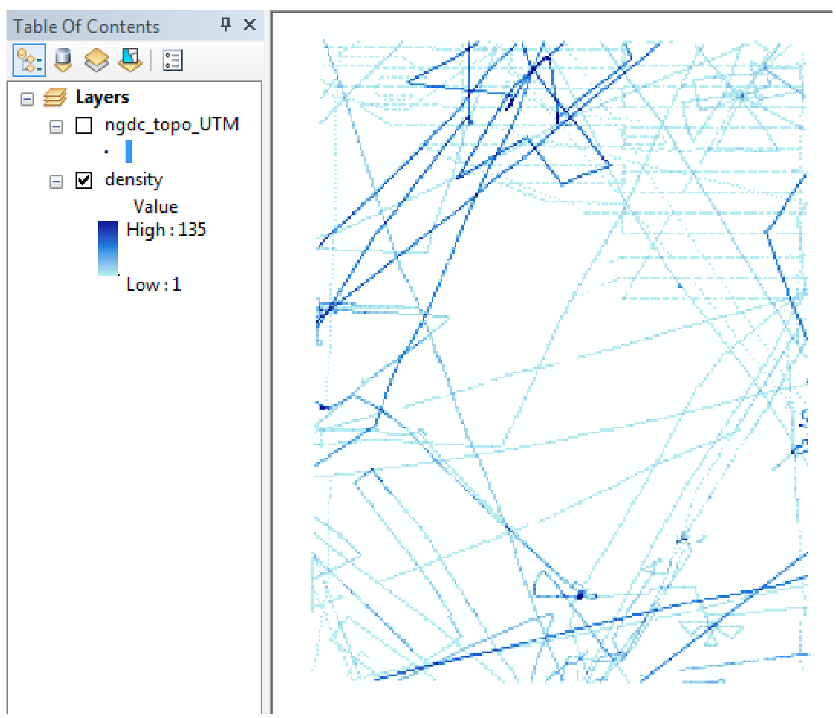


- Use your projected data set as Input Features. Leave default FID in Value field. Choose a name and directory for your output file. Choose COUNT as Cell assignment type. It will assign the number of points within the cell to the output raster's cell. Leave Priority field with NONE and type your gridding cell size in meters in Cellsize box.

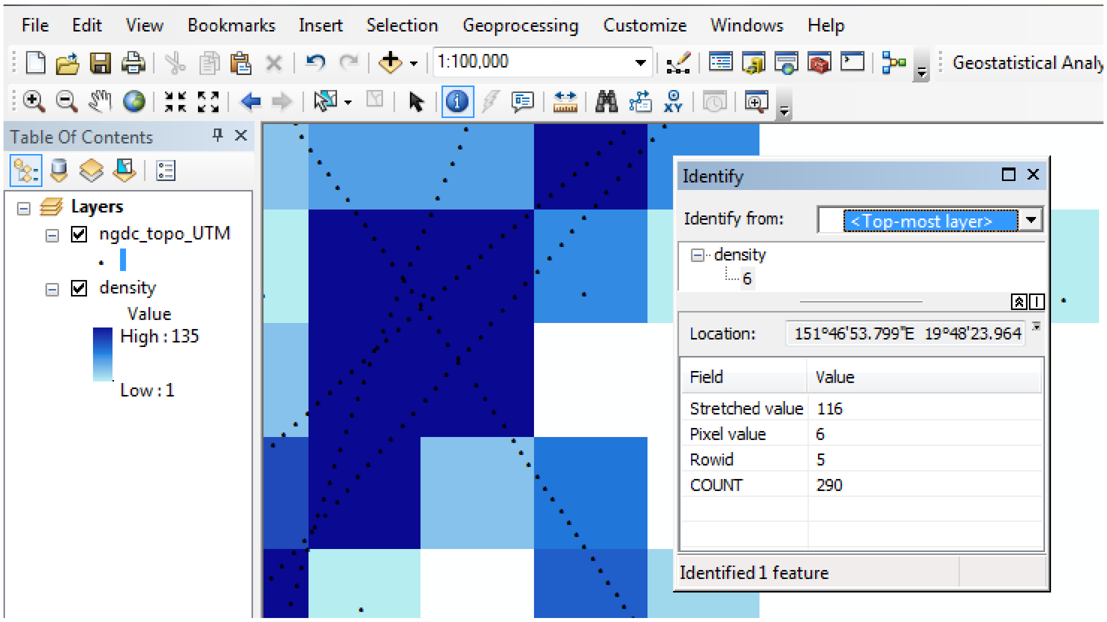


Find more information about Point to Raster tool here:
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Point_to_Raster/001200000002z000000/

- This is the result.



- You can zoom in and check the results of your computations. Shape file with your data points is displayed over the density raster layer. Use the “Identification tool” to click on a cell and check the number of data points within it. It will appear in Identify window under the name of layer and as a Pixel value. Moreover, as a Count value, you have a total number of cells with the same points number as chosen cell.



- Maximum density of data points for 2000 m grid size is 135 points. This grid size will be used in the further part of this example, but you may want to change it.

2.2.4 Creating a gridded surface

- There are several interpolation algorithms available in ArcMap, these include Global and Local Polynomials, Kriging, Spline, Natural Neighbor and Inverse Distance Weighting available within Spatial Analyst and 3D extensions. The description of some of these interpolation algorithms can be found in Chapter 8.0. More information on interpolation from point to surface methods can be found:

  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_the_Raster_Interpolation_toolset

  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_the_Interpolation_toolset

A list of interpolation algorithms are available through Spatial Analyst and Geostatistical Analyst toolboxes:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_introduction_to_interpolation_methods/003100000008000000/

- Here we will present how to create a gridded surface using Inverse Distance Weighting (IDW) algorithm using Geostatistical Analyst.

- IDW can be a good choice for fast interpolation of sparse data. IDW interpolation predicts values at unmeasured locations according to the values from the surrounding data points. Points which are closer to the prediction location have more influence (weight) on prediction than those which are further away. For more information on IDW see:

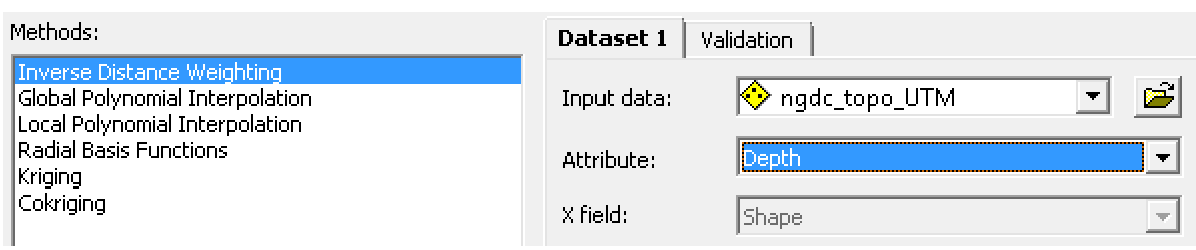
  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=3304&pid=3302&topicname=How_Inverse_Distance_Weighted_%28IDW%29_interpolation_works

- In order to choose optimal parameters for interpolation, you can use Geostatistical Analyst toolbox. Activate the extension through Tools> Extensions> check Geostatistical Analyst. Also add it to your display through View> Toolbars> Geostatistical Analyst.

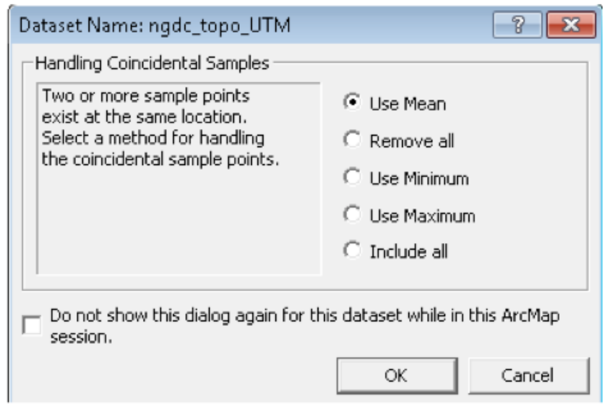
- Geostatistical Analyst> Geostatistical Wizard provides cross-validation of your model according to parameters chosen. The wizard allows you to manipulate parameters and to look at the output statistics for the created model. For more information see:

  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=3355&pid=3334&topicname=Performing_cross-validation_and_validation

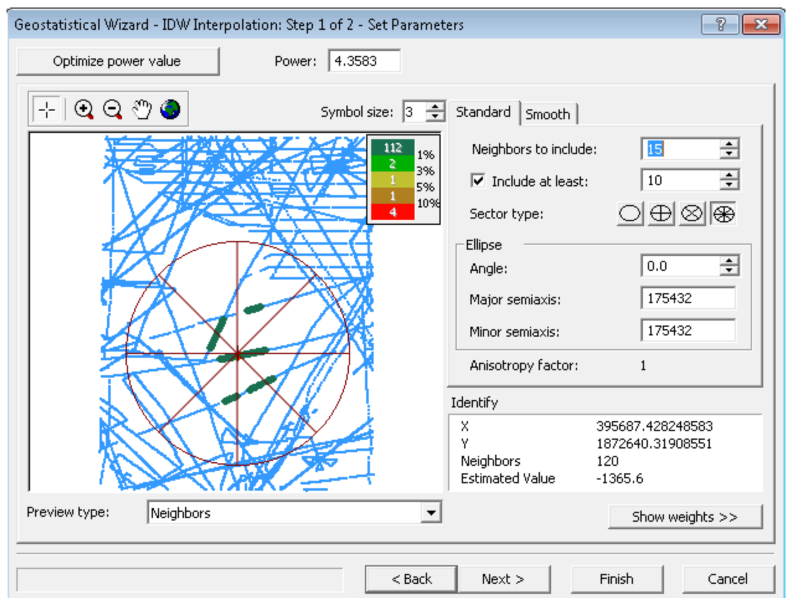
- In Geostatistical Analyst open Geostatistical Wizard. Select our input data point layer ngdc_topo_UTM and Depth as an attribute:



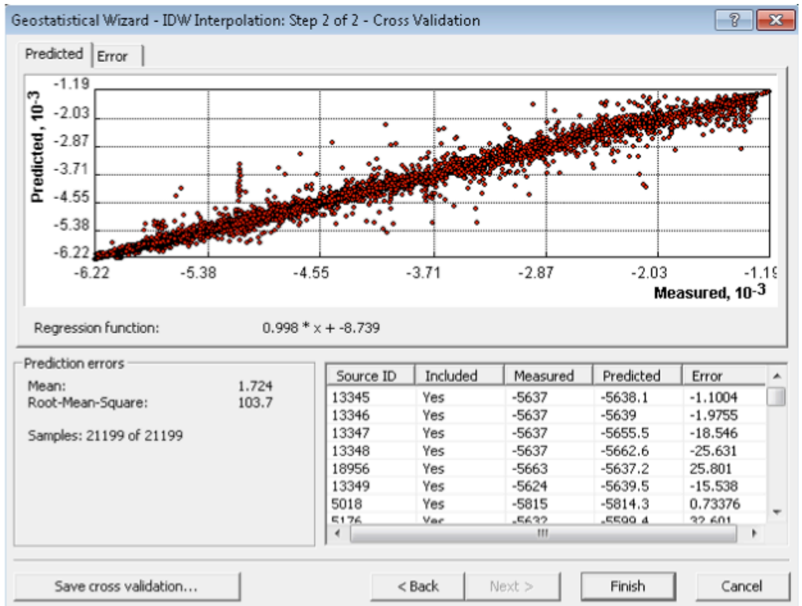
- Click next. In the appeared window select Use Mean> Ok



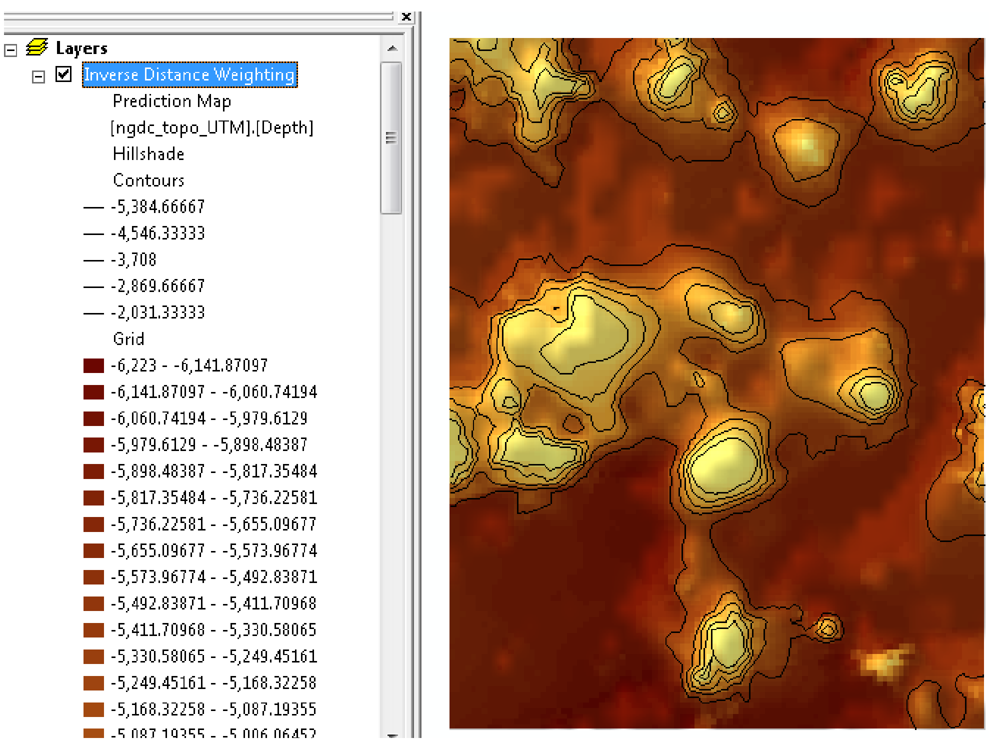
- A set Parameters window will appear. As an example we will use *Neighbors to include* 15, Include at least 10, sector type *Four* . After we set the parameters we can click Optimize power value button on top left corner. Optimized power will be 4.3. Also you can change the preview type to *Surface*, and while changing the parameters you can see how it will affect the output surface. Click next.



- The window of Cross-Validation will appear. The root mean square error is more than 100 meters. You can vary interpolation parameters and see how they affect on the model fit. Click Finish.



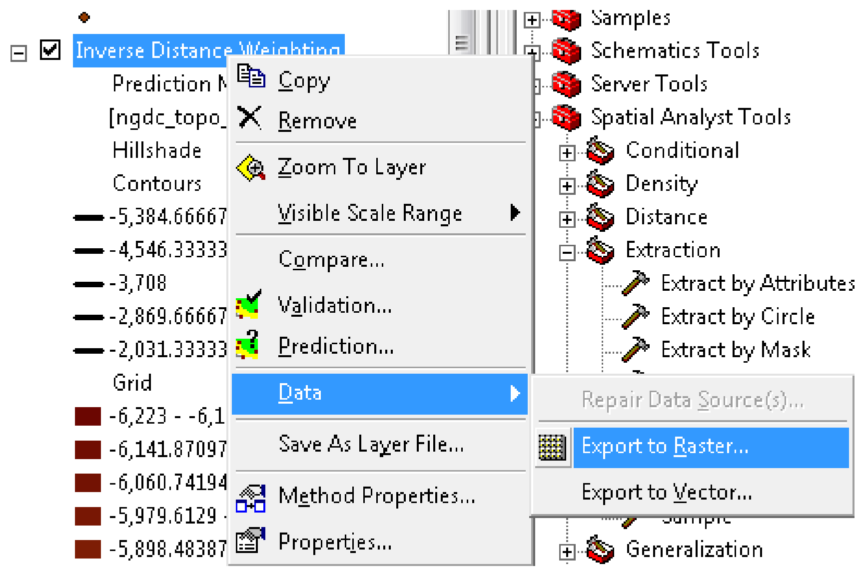
- The summary window will appear. Ok
- The surface is being created
- After adjusting the color bar and checking contours (Layer Properties>Symbology), our final surface using above described parameters looks like that:



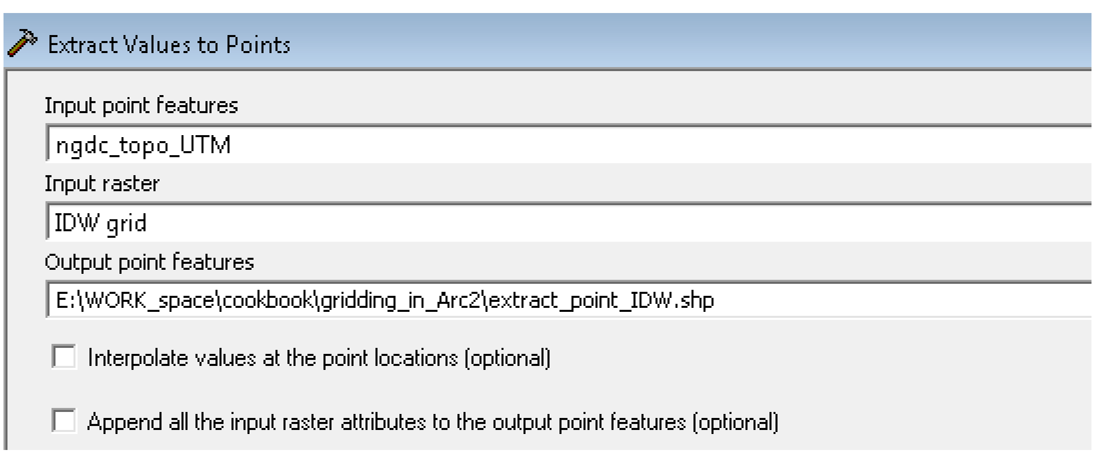
2.2.5 Comparison with original data values

- Here we present how to compare the original data point values with created surface. This could be useful in order to investigate spatial distribution of misfit between model and the measured points. For more information on quality assessment of grids see Chapter 4.1.

- Export our Inverse Distance Weighting layer as ESRI grid: Layer> Data> Export to Raster. Set cell size of 2000. Save as "IDW grid" file name

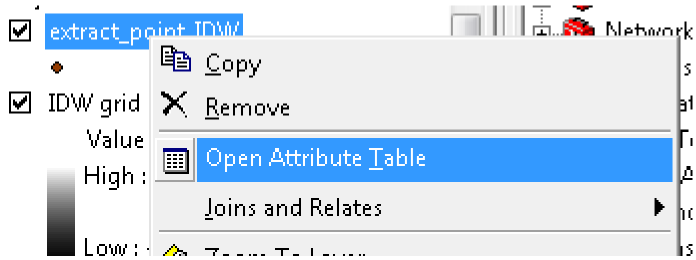


- Now we can extract values from the gridded surface at corresponding original data points with Spatial Analyst Tools> Extraction>Extract Values to Points. Where Input point features is our ngdc_topo_UTM points and Input raster is our IDW grid interpolated raster . OK

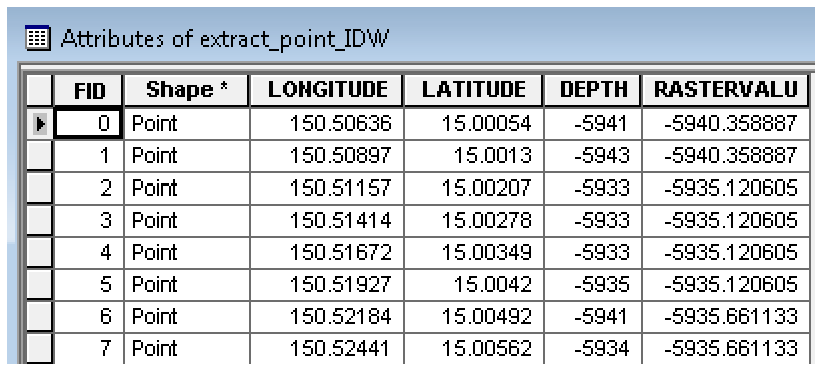


- As a result, a point feature layer is created (we call it extract_point _IDW).

- You can view the Attribute table of the layer by right click on the layer> Open attribute table:



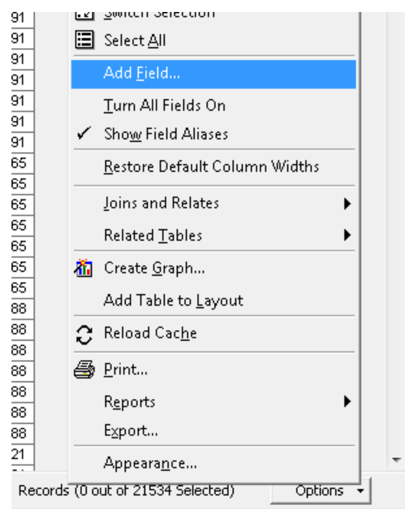
- As you can see from the table it contains information for original data points (x,y,z) and corresponding depth values from our gridded surface (RASTERVALU) for each original data point:



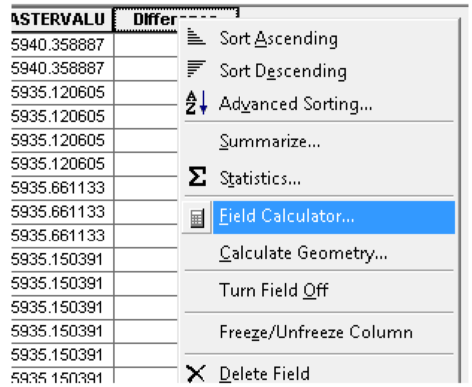
Attributes of extract_point_IDW

FID	Shape *	LONGITUDE	LATITUDE	DEPTH	RASTERVALU
0	Point	150.50636	15.00054	-5941	-5940.358887
1	Point	150.50897	15.0013	-5943	-5940.358887
2	Point	150.51157	15.00207	-5933	-5935.120605
3	Point	150.51414	15.00278	-5933	-5935.120605
4	Point	150.51672	15.00349	-5933	-5935.120605
5	Point	150.51927	15.0042	-5935	-5935.120605
6	Point	150.52184	15.00492	-5941	-5935.661133
7	Point	150.52441	15.00562	-5934	-5935.661133

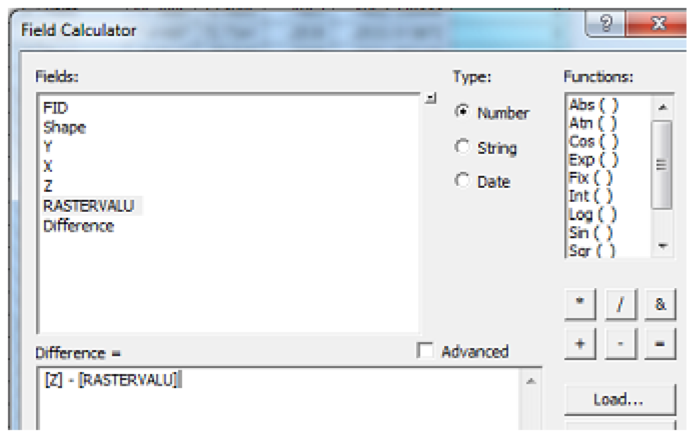
- You can calculate differences between these values and visualize them. Go to Options on inside the Attribute table (bottom right corner) and select Add Field option. We will call the new column as "Difference".



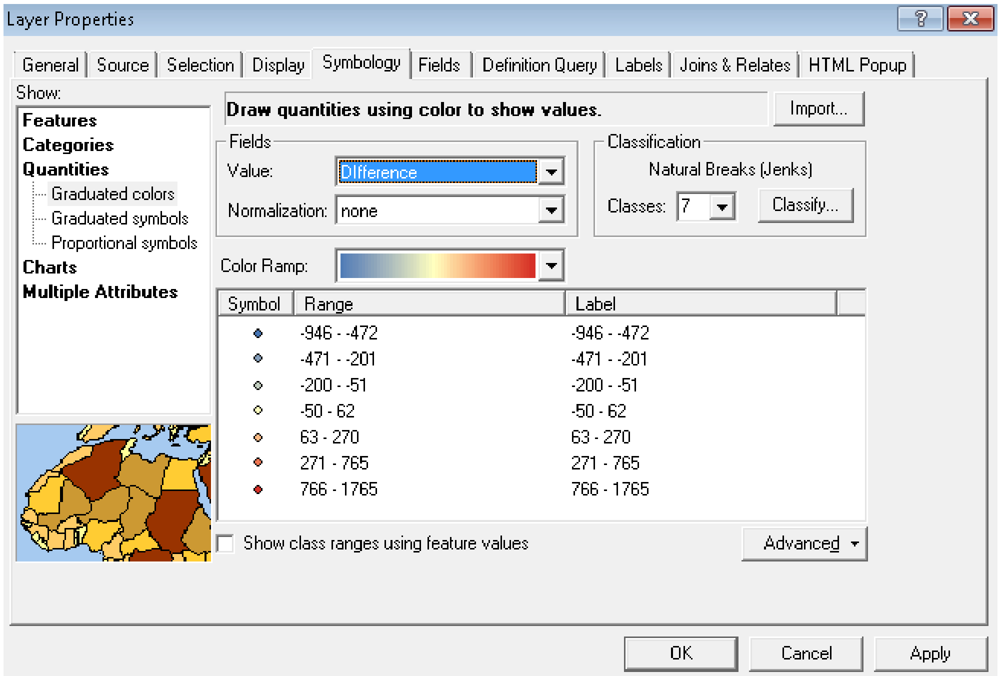
- After the column is created, right click on the header of new created column and select Field Calculator



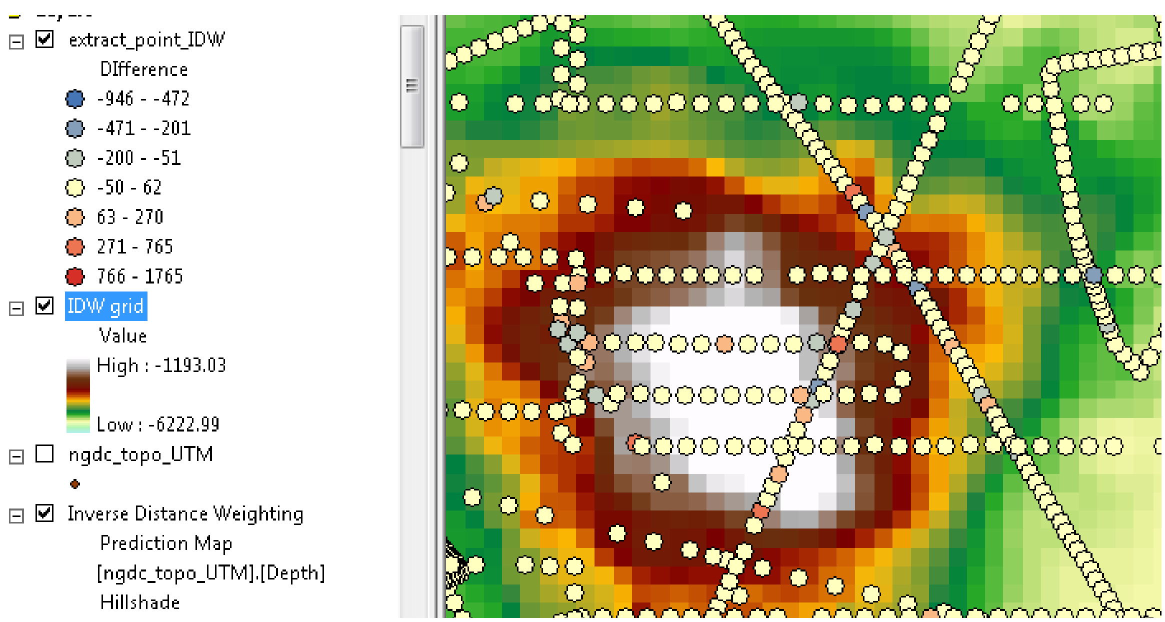
- Inside the Field calculator we can define expression for the new column values: a difference between Depth and RASTERVALUE columns (double click on the Fields to add them to the expression). After clicking Ok, the values are calculated. Close the attribute table



- Now you can visualize the Difference by right clicking on the point layer (Extract_point_IDW)> Properties. Go to Symbology tab. Select Quantities> Graduated colors. Select the Value Fields as Difference, select the color ramp and number of classes, and click Ok.



- Now you can see overlaid difference results over the created gridded surface and investigate the regions where differences seem to be significant (make Layers" IDW grid" and "extract_point_IDW " visible):



- For more information on analyzing gridded surfaces that you create go to:

  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=About_analyzing_raster_data

  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=603&pid=598&topicname=Surface_creation_and_analysis

See more on interpolation error analyses:

  http://webhelp.esri.com/arcgiSDEsktop/9.3/index.cfm?TopicName=About_geoprocessing_with_3D_Analyst

FUNDAMENTALS

Chapter 3.0 Gathering Software

Some software packages are freely available for download from the internet. The purpose of this chapter is to show where to find such software and how to obtain it.

3.1 Generic Mapping Tools

Contributed by K. M. Marks, NOAA Laboratory for Satellite Altimetry, USA

GMT (Generic Mapping Tools) (Wessel and Smith, 1998) is a collection of open source mathematical and mapping routines for use on gridded data sets, data series, and arbitrarily located data. The GMT package is available for download from the University of Hawaii website (http://gmt.soest.hawaii.edu/) (see Figure 3.1). We utilized GMT routines for much of our data analyses, gridding, and mapping discussed in selected chapters that follow. Software packages such as MATLAB, IMSL, ArcGIS and others may also provide similar mathematical and mapping capabilities, but these are not free.

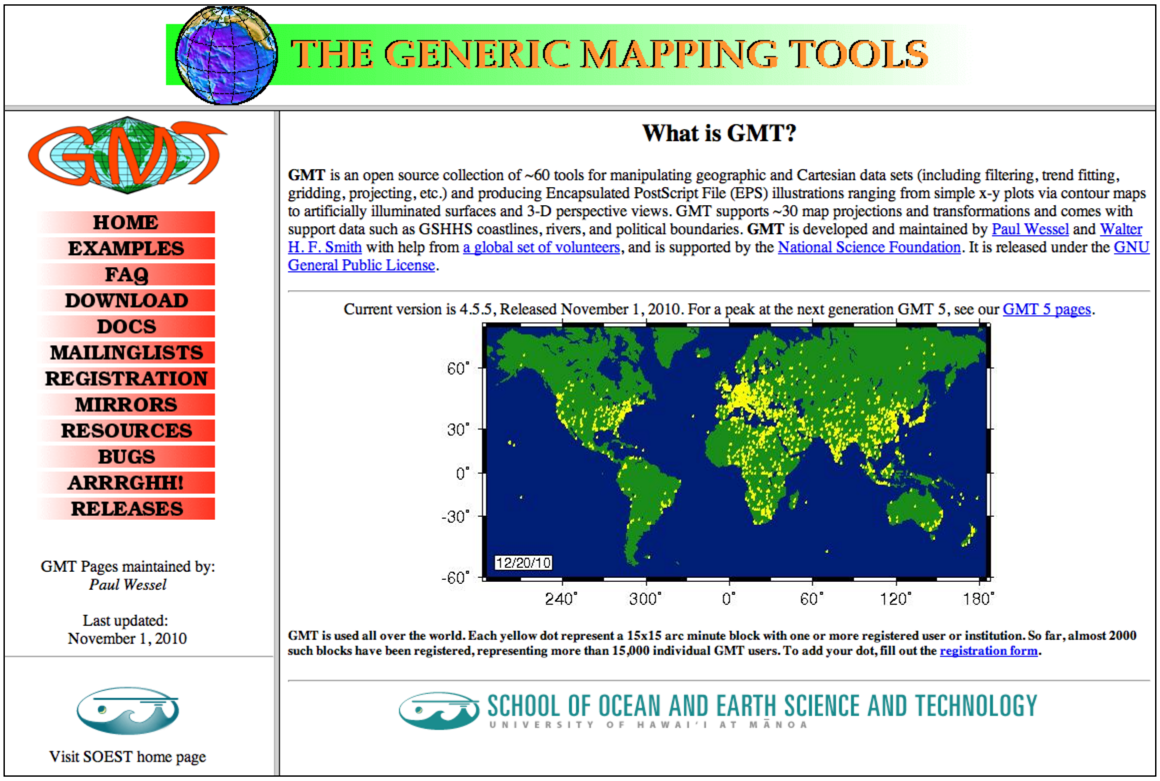


Figure 3.1 University of Hawaii website for GMT.

To download the GMT software package, click on "download" shown in the left column of the figure above. Instructions for how to obtain and install the GMT software are given on the download webpage (see Figure 3.2).

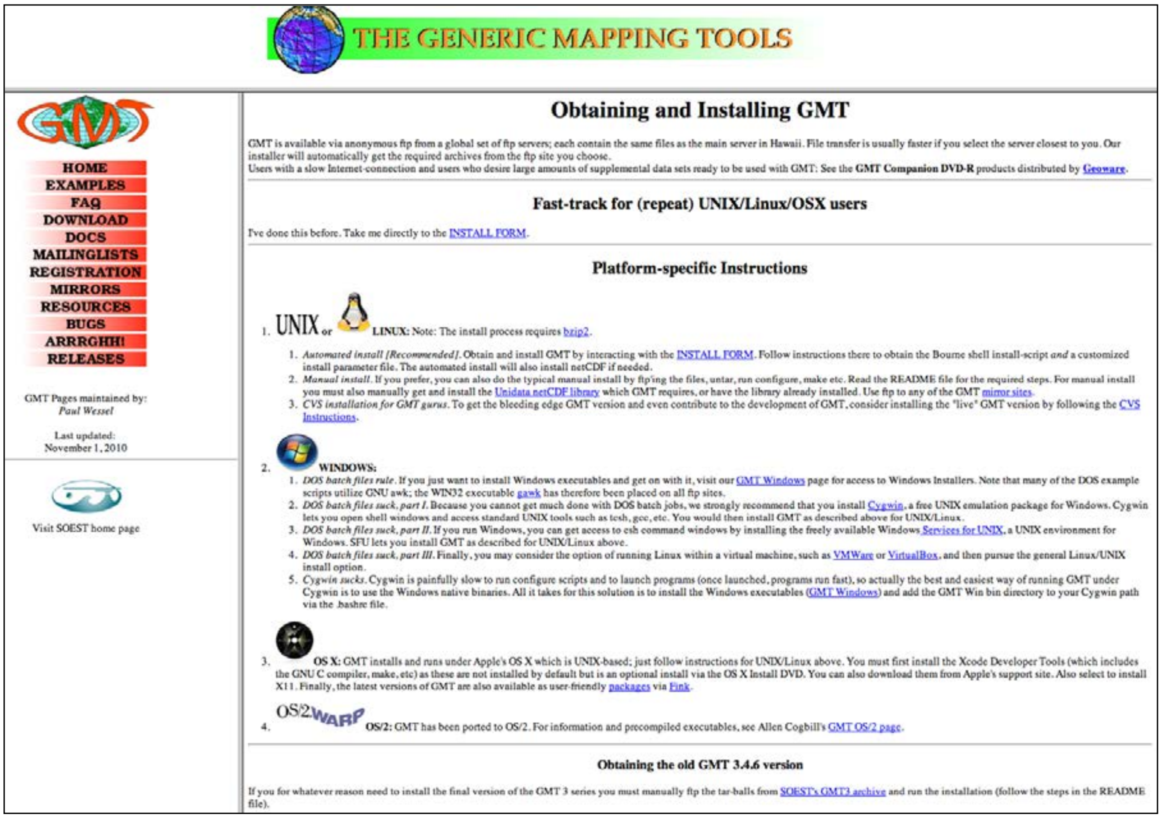


Figure 3.2 GMT download and installation web page.

Chapter 4.0 Software Overview

The purpose of this chapter is to give an overview of various software packages that can be used to clean, process, manipulate, analyze, and plot data used in scientific research and production.

4.1 ArcGIS

Contributed by Pauline Weatherall, British Oceanographic Data Centre (BODC), UK

The software company, ESRI (www.esri.com), produces a range of Geographic Information System (GIS) products suitable for accessing, processing, sharing and serving geo-referenced data, working both in a MS Windows environment and also via the internet.

Further information about ESRI's products; accessible data sets; development tools; specialized applications and open source products can be found on ESRI's web site:

http://www.esri.com/products/index.html

The information below aims to give a very broad overview of some of the functionality of A**rcGIS for Desktop software**, specifically in relation to GEBCO's work with building gridded bathymetric data sets. The information and links below relate to ArcGIS Desktop version 10.

4.1.1 What is ArcGIS for Desktop Software?

ArcGIS for Desktop is ESRI's MS Windows-based GIS software. As part of its functionality, the software can be used to:

- View, overlay, manage, edit, analyse and capture geo-referenced data
- Produce and share maps, data and imagery
- Grid data
- Search for data and imagery
- Carry out datum and projection transformations
- Edit/develop metadata

4.1.2 Is the software free, what licences are available and how can you access the software?

ESRI makes available a range of desktop software products, some are available free of charge and some are available commercially at a number of **licensing levels**. The functionality of the software depends on the licensing level and is also reflected in the price of the software.

Further details about software access and licensing for each product can be found on ESRI's web site:

http://www.esri.com/products/index.html#desktop_gis_panel

4.1.3 What type of computer platform does it run on?

ArcGIS for Desktop products are designed for use in a MS Windows environment (which includes Intel-chipped Macintosh computers that have a windows emulator). Details about supported operating systems and hardware and software requirements can be found on ESRI's web site:

http://resources.arcgis.com/content/arcgisdesktop/10.0/arcgis-desktop-system-requirements

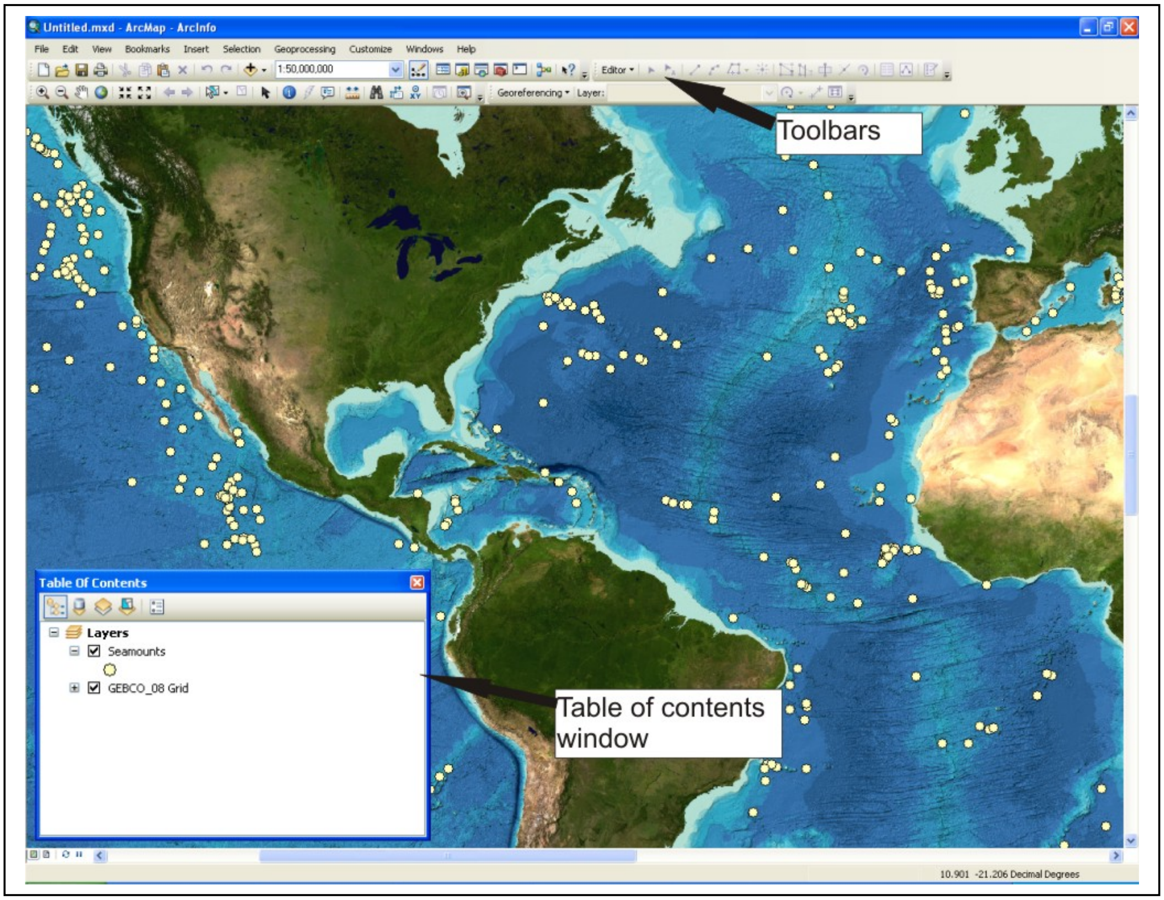
4.1.4 What does the software do?

As outlined above, the software can be used to visualise, edit, capture, process, share and analyse geo-referenced data.

Firstly, we need to look at what the various component packages of the ArcGIS Desktop products do and what types of data and data formats they can work with. The following is a broad overview of their functionality, further information can be found in the products' documentation.

ArcGIS for Desktop application packages - what are they and what do they do?

ArcMap is used for creating, displaying, exploring, analysing and editing your data; assigning styling to the display and for building images and map layouts from individual data layers or data sets.



Displaying information from GEBCO's data sets in ArcMap – a Tiff image overlain by a shapefile of point data

An introduction to ArcMap:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//006600000001000000.htm

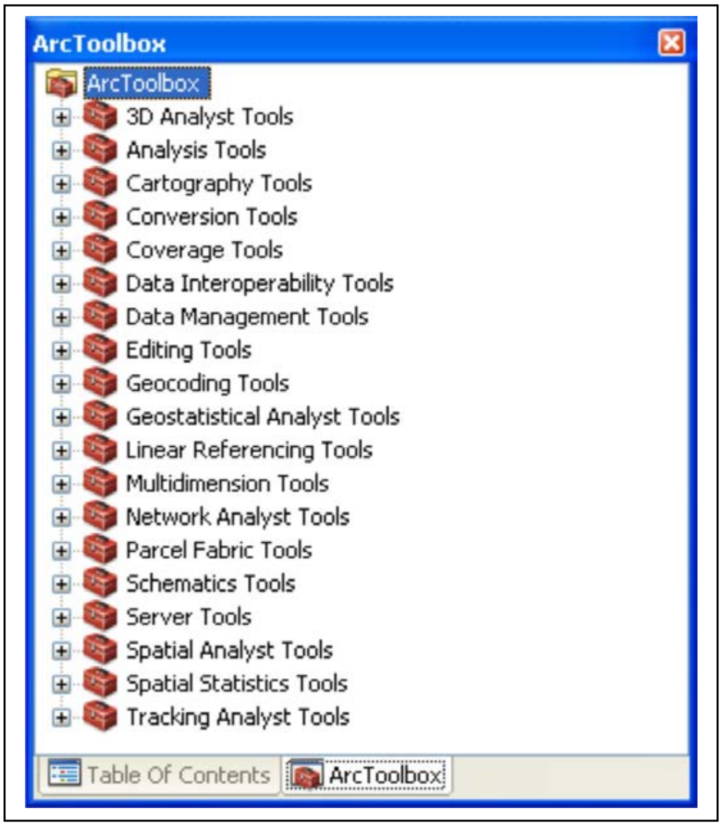
A quick tour of ArcMap:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/A_quick_tour_of_ArcMap/006600000332000000/

ArcCatalog lets you organise, preview and manage your data sets. In simple terms it can be thought of as acting a little like MS Windows Explorer in that it provides an overview of the contents of the directories on your PC. However, it displays the information in a form that is useful for working with ESRI software, i.e. although a shapefile is made up of at least three separate files when viewed with Windows Explorer, within ArcCatalog it is just represented as one file. ArcCatalog also lets you view and enter metadata for your data sets in a number of standard formats. A catalog window is available in ArcMap to let you manage your data sets.

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//006m00000069000000.htm

ArcToolbox is a collection of data processing tools collected into related ‘toolboxes’ and ‘toolsets’. They have a wide range of functionality, including data analysis and conversion tools. The availability of the various toolboxes and toolsets depends on the level of your ArcGIS Desktop license.



Some of the Tool sets available through ArcToolbox

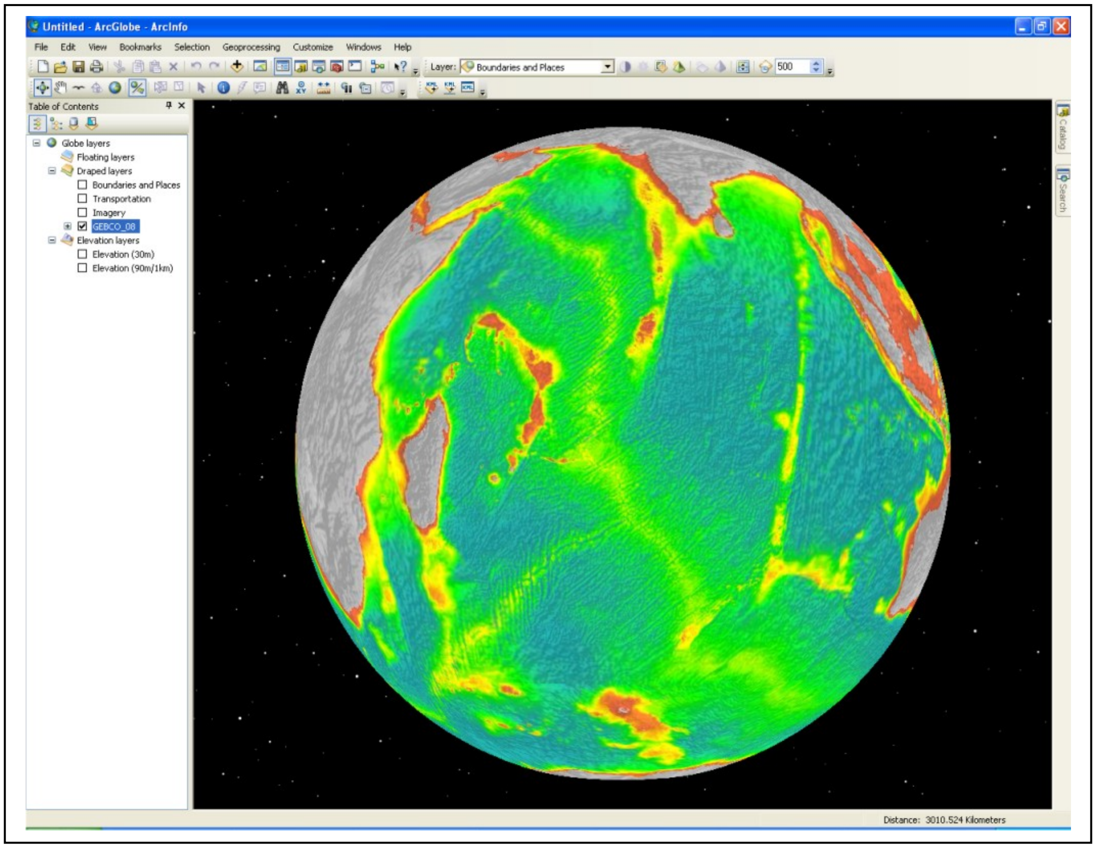
A quick tour of geoprocessing tools:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/A_quick_tour_of_geoprocessing/002s00000002000000/

Information on finding tools can be found at:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/A_quick_tour_of_finding_tools/002400000001000000/

The packages, **ArcScene** and **ArcGlobe** are part of the 3D Analyst extension package and can be used to view data in 3D and produce and animations (ArcScene) and display data on a globe (ArcGlobe).



Displaying the GEBCO_08 Grid in ArcGlobe

Further information about extension packages can be found on ESRI’s web site:

http://www.esri.com/software/arcgis/arcgis-for-desktop/extensions.html

In what formats and structures does ESRI ArcGIS Desktop software store data?

ESRI ArcGIS Desktop software works with and stores data in a number of ways, as individual data files or as collections of related data files. The data sets may be stored individually on disc or be part of a database. The data sets can be in the form of 3D surfaces (e.g. rasters), vectors (e.g. points, lines or polygons) or images.

The following outlines the main data storage types used by ESRI’s ArcGIS Desktop software products.

Geodatabase

This is the main data model for ArcGIS and is the primary data format used for editing and data management. It is made up of a collection of files (feature classes, raster data sets and tables). **Feature classes** are collections of features of the same type, such as points, lines or polygons. They also share a common set of attribute fields. Within a Geodatabase they can be grouped into

feature data sets which are collections of feature classes with the same geographic reference information (i.e. datum and projection).

Geodatabases work across a range of database management system (DBMS) architectures and file systems, can be of various sizes, and be single or multi-user.

Within some ArcDesktop GIS packages, storing your data in a geodatabase allows you to carry out advance editing and quality control procedures on vector data sets, i.e. looking at how the vector data sets which make up the geodatabase 'interact' or share geometry – this is called topology. For example making sure the outlines of individual polygons do not overlap. This functionality is not available for shapefiles.

Further information about geodatabases can be found on ESRI's web site:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//003n00000002000000

Shapefiles

The Shapefile is a format that ESRI software uses to store the shape, geometric location and attribute information about a data set. Individual shapefiles can contain only one feature 'type', i.e. points, lines or polygons. Shapefiles can be edited but they cannot be used as part of a topology for more advanced quality control procedures.

Further information about shapefiles can be found on ESRI's web site:

http://downloads2.esri.com/support/whitepapers/mo_/shapefile.pdf

3D Surfaces

ESRI products work with the concept of three types of surface models:

- **Rasters –** rectangular array of grid cells (holding a data value) arranged in rows and columns

- **Triangular Irregular Networks (TINs)** - a surface consisting of nodes (or points) joined by edges into various sizes of triangles. Because the nodes can be irregularly spaced, a TIN can show higher resolution information where there is more variation in terrain and be lower resolution where there is less variation.

- **Terrain surfaces** – are a set of 'pyramided' TINs, i.e. a set of TIN's at multiple levels of resolution. This helps when working with high volume data sets.

Find out more about how ArcGIS software works with 3D surfaces:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Fundamentals_of_Surfaces/00q80000005z000000/

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00q8000000rm000000

How do you work with your data in ArcGIS Desktop packages – do you have to reformat it?

As described above, ArcGIS Desktop packages work with data in a number of forms, as individual 'stand-alone' files or as part of a geodatabase.

Data in some formats can be imported or used directly with ArcGIS Desktop applications, others need to be converted to ESRI formats.

Information about supported data formats for import and export can be found on ESRI's web site:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//009t0000000q000000

Through the **ArcGIS Data Interoperability Extension** tool set, data can be imported, exported and converted to a number of formats. Further information can be found at:

http://www.esri.com/software/arcgis/extensions/datainteroperability/index.html

Further information about importing data is given below:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_To_Raster_toolset/00120000002r000000/

In addition to creating rasters from input point data sets, you can also create rasters from polyline and polygon data sets. Further information is available at the above link.

Connecting to Open Geospatial Consortium (OGC) Web Services

You can connect directly to Web Map Services (WMS), Web Coverage Services (WCS) and Web Feature Services (WFS). Further information can be found on ESRI's web site:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//003700000002000000.htm

Working with KML files

Information on working with KML files can be found on ESRI's web site at:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00s20000000m000000.htm

Adding X Y data

Point data, in ASCII format, can be imported into ArcMap as a layer using the **Add XY** option, attribute information can also be imported along with the data. The data can then be exported as a shapefile or feature class.

Working with netCDF files

Grid files, in 2D netCDF form, can be imported into ArcMap as a 'netCDF raster layer' through the 'Make NetCDF Raster Layer' Multidimension Tools set in ArcToolbox. Further details can be found on ESRI's web site:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//004300000006000000.htm

Additional tools for data import

There are 'add on' tools such as ETGeowizards (http://www.ian-ko.com/ET_GeoWizards/gw_main.htm) which can be used to help with the import and export of vector and raster data sets for use with ArcMap.

Can you create a 3D surface from your source data sets?

The following outlines how you can use ESRI's software to create a raster or Triangular Irregular Networks (TIN) from your input data.

Creating a raster from point data

As part of the ArcGIS 3D Analyst extension software package you can create a raster from a data set of points using a number of interpolation methods:

- Inverse distance weighted
- Spline
- Kriging
- Natural neighbours

Further information about creating rasters can be found in the ArcGIS 3D Analyst product documentation, available from ESRI's web site:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00q900000034000000

Further information about gridding using ESRI software can be found in the section 8.2.10 of this document, 'Gridding with ArcMap'.

Working with raster surfaces

The following are examples of some simple tasks that you may want to carry out, for the full range of functionality see the ArcGIS Desktop documentation.

Analysing surfaces and generating contours

You can analyse the surface of your 3D model, looking at slope and aspect and hill shade and also **generate contours** at specified intervals. This can be done through the 3D Analyst extension – which provides a set of tools for working with rasters, TINs and terrain surfaces. Further information is given in the documentation.

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_3D_Analyst_toolbox/00q900000070000000/

The Spatial Analyst extension provides a set of tools for working with and analysing both raster and vector data sets.

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_Spatial_Analyst_toolbox/009z00000003000000/

Raster calculations

You can carry out mathematical operations on cells in a raster. This includes carrying out calculations on grid cells such as add, subtract or divide. This can be done by either adding or subtracting one grid from another or by adding or subtracting single values for each grid cell.

The following table lists the available toolsets and gives a brief description of each:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_Raster_Math_toolset/00q900000063000000/

Extracting data from rasters

You can extract data from rasters either based on geographic area 'clipping' or by cell values (attributes). For example, you may select a geographic sub-area from a grid or select cells from a grid with a depth greater than say 100m.

The majority of the extraction tools are available with the **Spatial Analyst Extension**; however, the 'Clip' tool is available with any ArcGIS license.

Further information about the 'Clip' tool can be found at:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00170000009n000000

For information about data extraction as part of the spatial analyst extension see:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_Extraction_tools/009z00000028000000/

Resampling, generalising and filtering rasters

A set of resampling, generalisation and filtering tools is provided as part of the ArcGIS Desktop products.

Resampling a raster, i.e. changing the size of the grid cells can be done through the 'resample' tool, part of the Data Management Toolbox:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Resample/00170000009t000000/

The ArcGIS Spatial Analyst extension contains a number of tools for generalising and filtering rasters:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_Generalization_tools/009z00000033000000/

Working with your data in ArcMap

ArcMap is the ArcGIS Desktop package which can be used to view and edit your data sets in 2D.

Further information can be found on ESRI's web site:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//006600000001000000.htm

A quick tour of the functionality of ArcMap is given at the following link:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/A_quick_tour_of_ArcMap/006600000332000000/

The following are examples of some of simple functions that you might want to do in ArcMap.

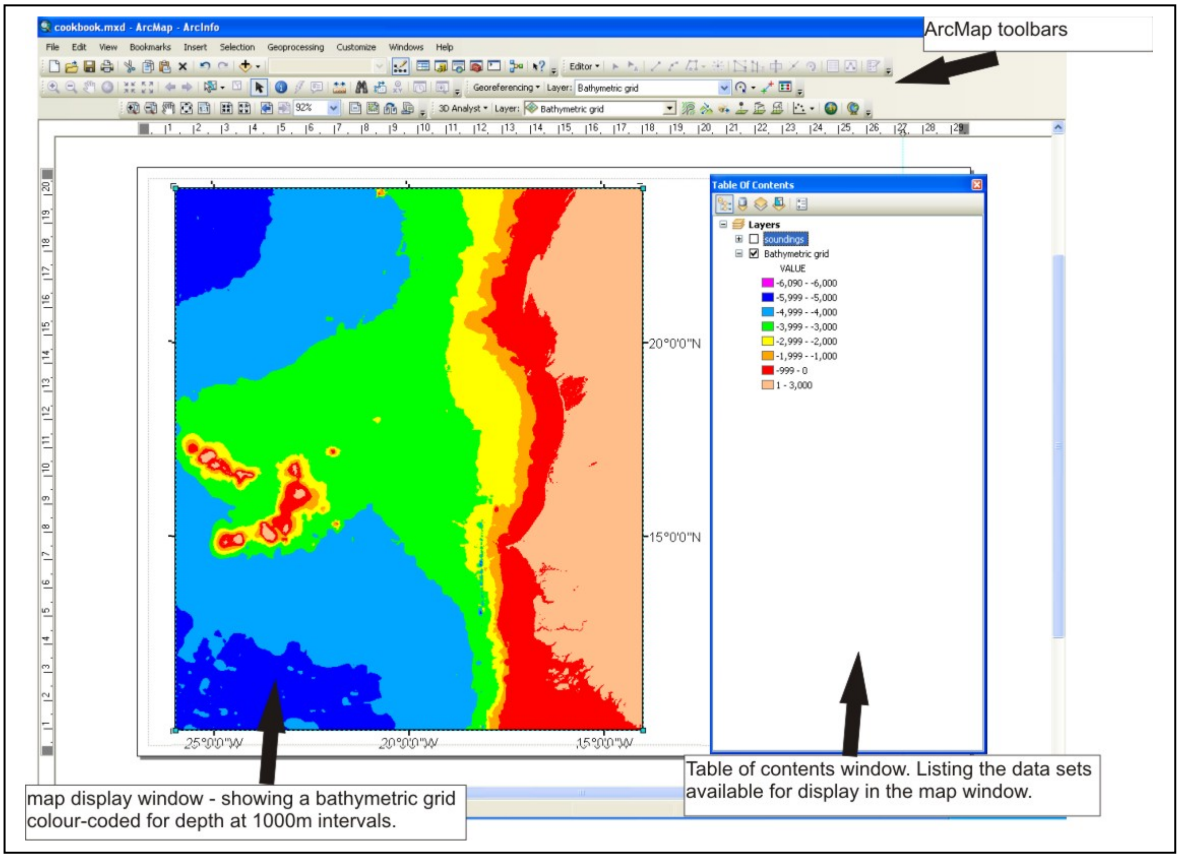
To view a data set, it needs to be in a format which can be recognised by ESRI ArcGIS Desktop software (see above for more information). Click on the 'Add Data' button from the ArcMap toolbar, then search your data directories to select the file that you want to load.

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Adding_layers_to_a_map/00660000000t000000/

In version 10 of ArcGIS Desktop you can also search for GIS content in your own file system and on the internet:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Using_search_in_ArcGIS/00660000007q000000/

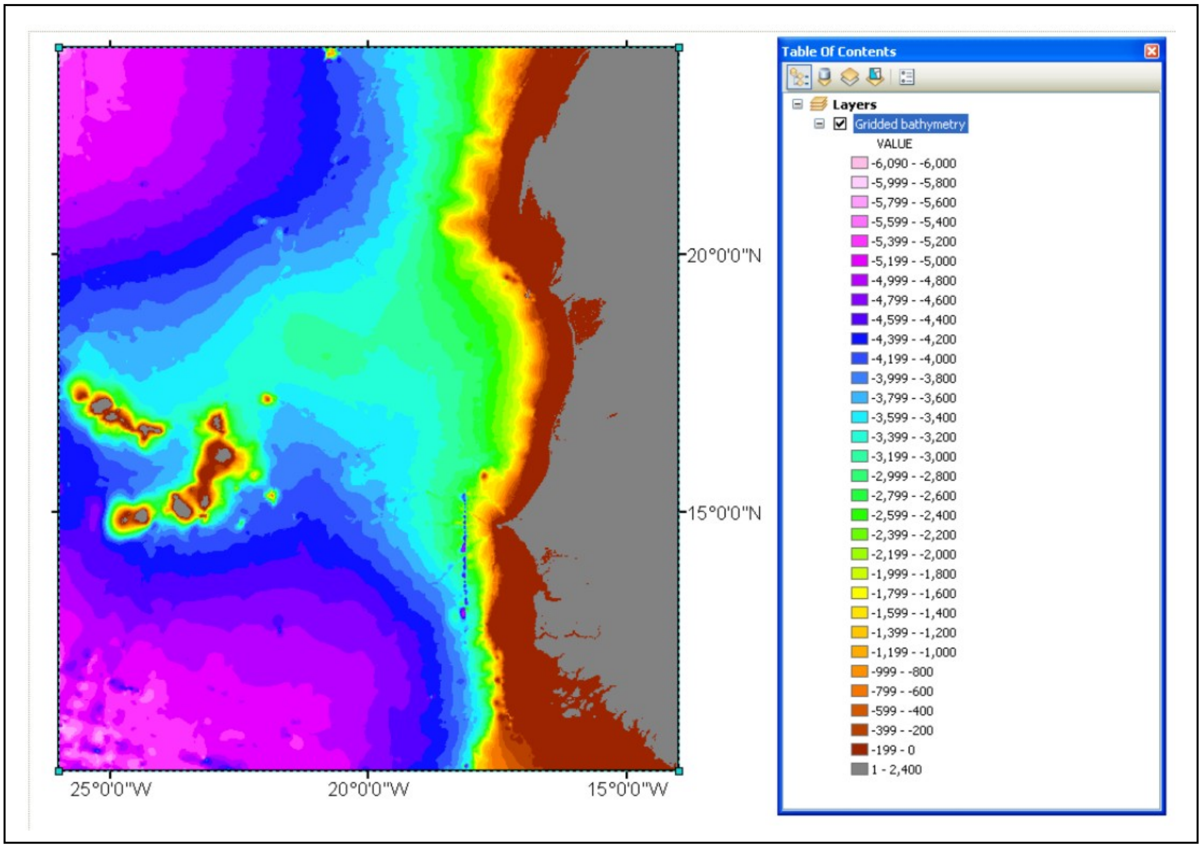
Once a data set is loaded you can experiment with changing how the displayed layer looks, for example, altering the colours or symbols used to display the data and adding label and text information. This can help to highlight aspects of the data set that you want to enhance, e.g. displaying data in a particular depth range all one colour.



In the above example a bathymetric grid is displayed colour-coded at 1000m intervals

The grid will appear in the layers list in the table of content. Right click on the grid's file name and then click on 'properties'. For raster data sets, you can choose to display the grid using stretched values along a colour ramp, you define the start and end colour values, using values and classified intervals, e.g. equal intervals, natural breaks, defined intervals etc. or unique values for each separate cell attribute. You can define a colour scheme, choosing from pre-existing colour schemes or defining and saving your own colour scheme.

You can change the properties of individual colour intervals, perhaps changing one interval to be transparent, this helps with overlaying data sets. You can also make a selected range of values one colour, for example for land areas.



In the above image the grid is now coded at depths of 200m intervals and land areas are displayed in grey.

Further information about choosing display colours can be found in 'help' system.

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/About_displaying_layers/00s50000001s000000/

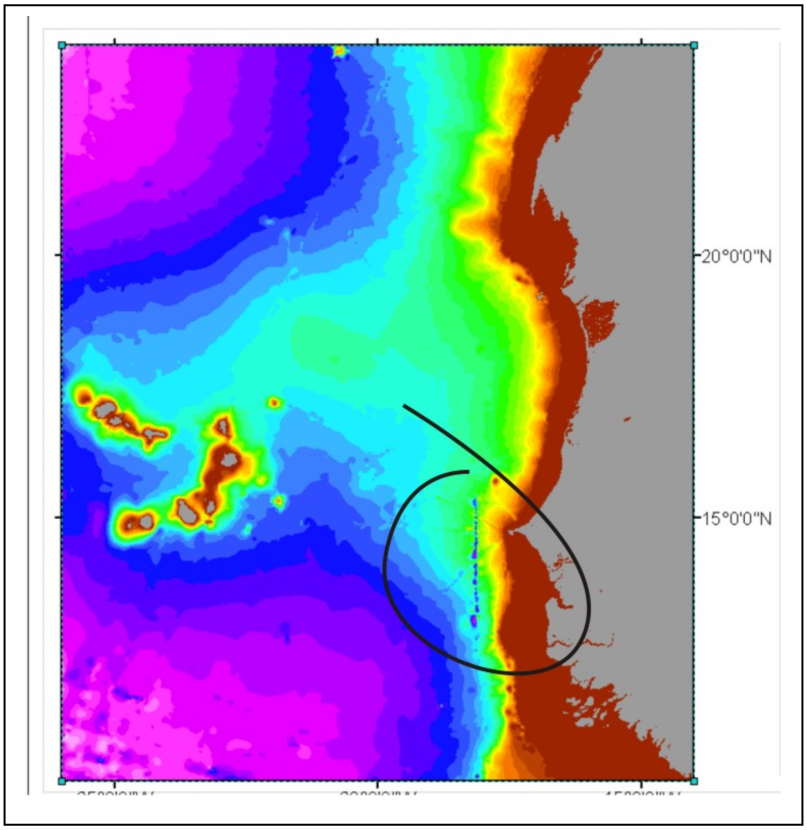
For vector data, you can change the symbology used to display the data set to get the most out of your data set. Symbols can be defined on size of attributes and/or colour and attribute information can be displayed on the map. For example, when plotting sounding points you can colour code them for attribute information such as depth or for contributing organisation.

You can change the drawing order of the displayed layers, for example to plot points on top of a map. For further information on working with layers:

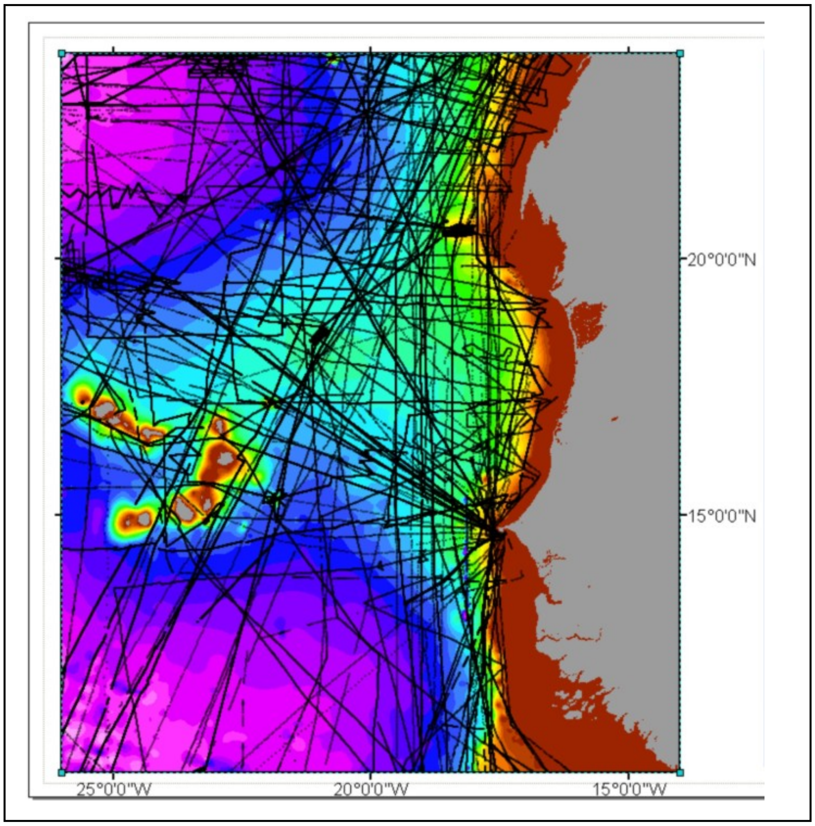
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/A_quick_tour_of_map_layers/00s500000015000000/

Example of using ArcMap and ArcScene to help assess a bathymetric grid

As described above, ArcMap can be used to visualise your data sets in 2D. In the following example a grid is loaded into ArcMap. Colour coding the grid at defined depth intervals highlights some features in the grid. Overlaying the source sounding data, used to generate the grid, can help investigate the origin of the features and let us determine if there are any errors in the source data.

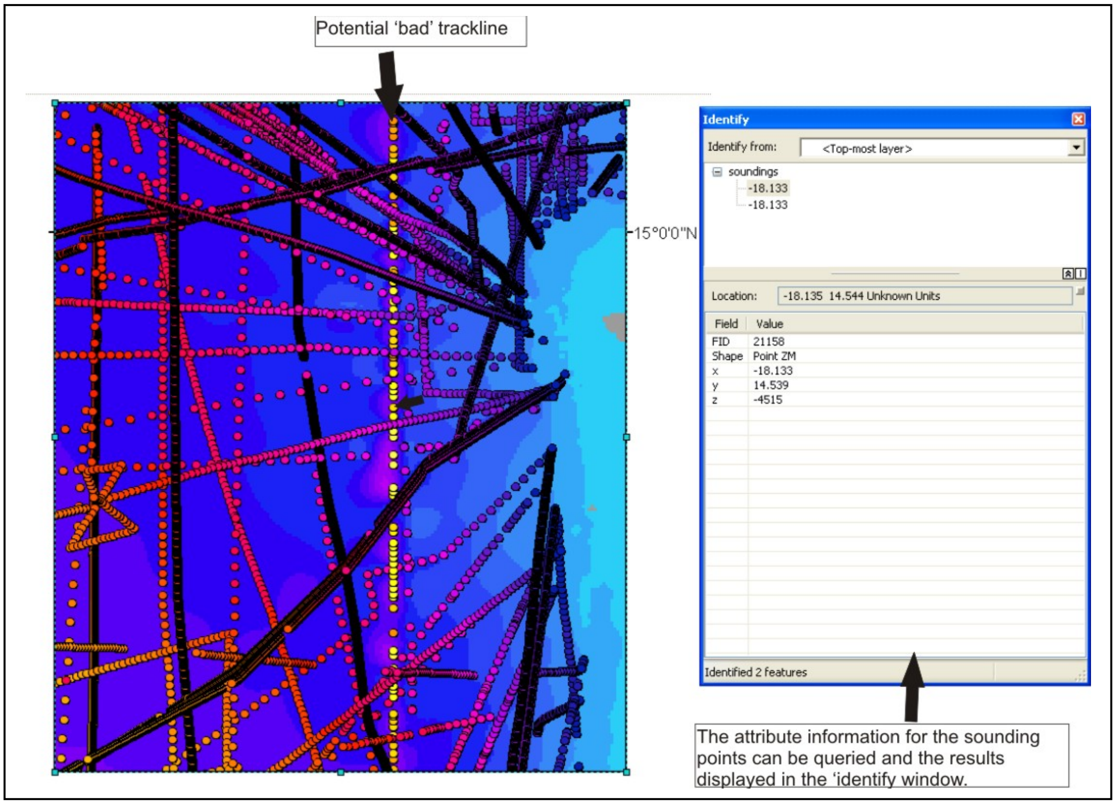


The above image seems to show an anomalous feature (circled), shown as a ‘blue line’. We can investigate the original of the feature by overlaying the source sounding data that was used to generate the grid onto the map display.



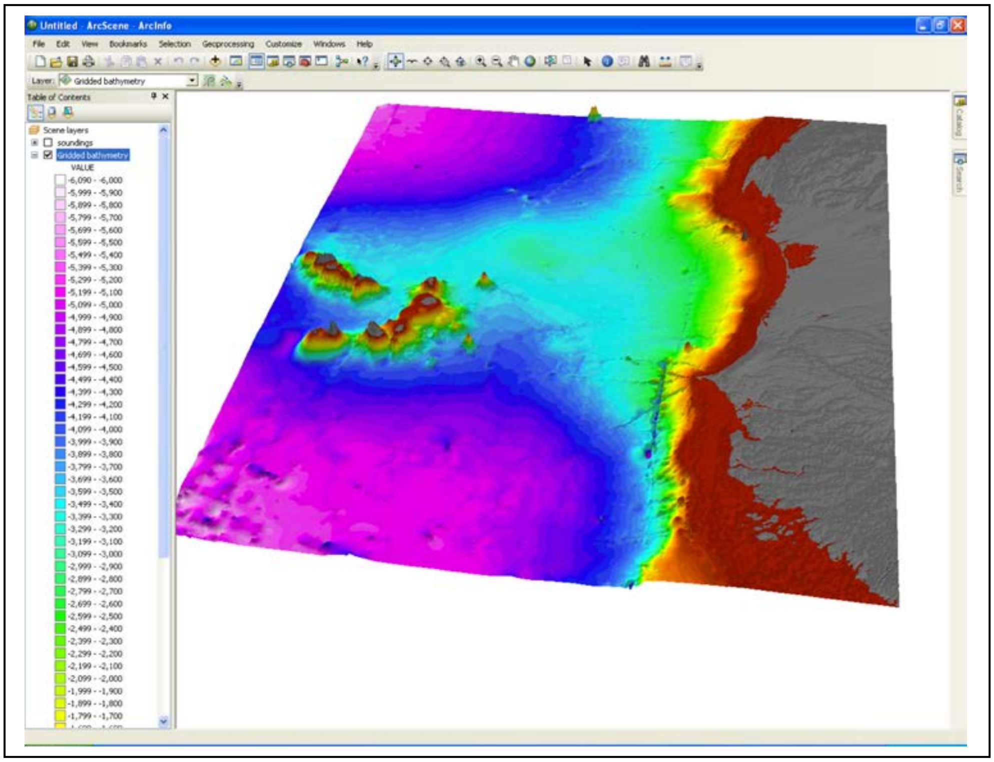
The above image shows the coverage of ship-track data (shown in black) which was used to generate the bathymetric grid.

The ship-track soundings, as xyz data, can be colour-coded for their various available attribute, e.g. depth or source survey. This can help to visually identify any spikes in the data set.



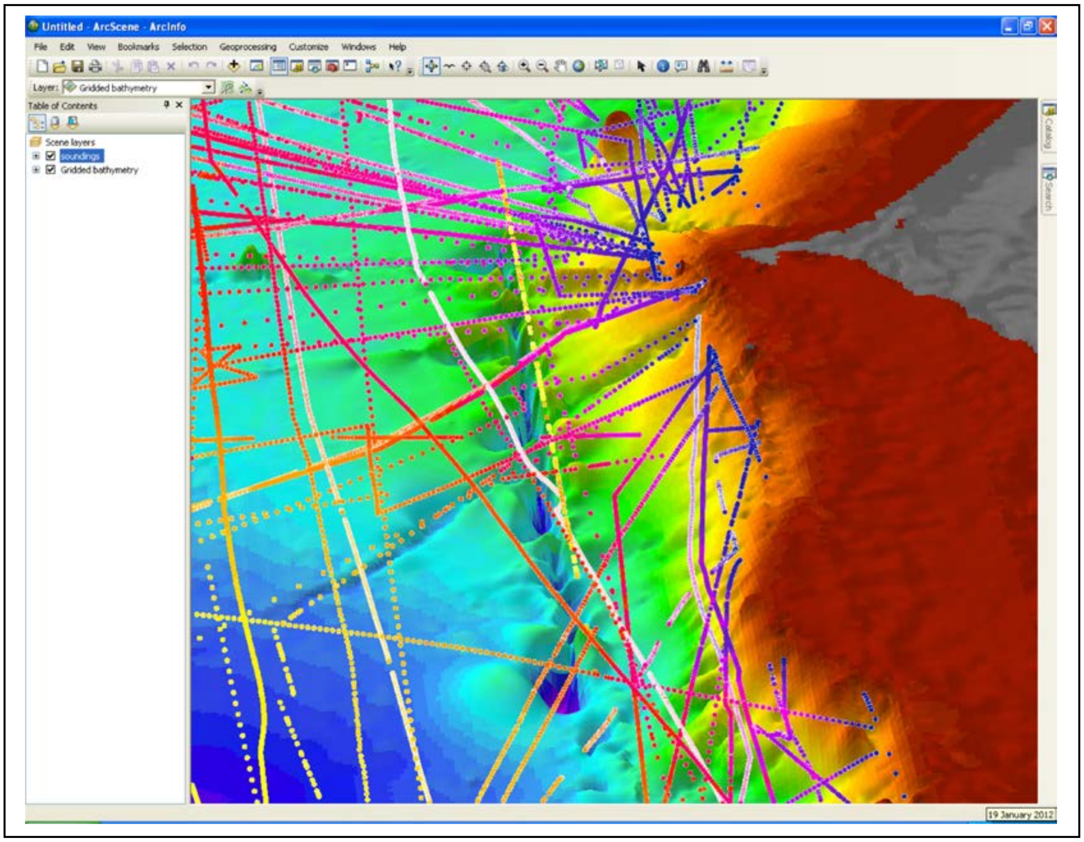
The above image is zoomed into the area of the anomalous feature. The source sounding data file has been colour coded for depth. We can see that a trackline plots along the anomalous feature, judging from the colour of the soundings, they appear to be a different depth value of those in the surrounding region – they may be in error and causing the anomalous feature in the grid. We can query the soundings to find out the depth of the individual sounding points and if available, query the accompanying metadata.

In addition to viewing the gridded data in 2D using ArcMap, **ArcScene** can be used to view the data in 3D.



The above image shows the grid displayed in 3D in ArcScene. Potential errors in the grid are now more visible.

We can zoom into the image to investigate further and also overlay the source sounding data onto the grid. In the example below, the anomalous feature is displayed in 3D and overlain by the source sounding data points.



How can I get help with using the software?

There are documents and links on ESRI's web site under 'support' http://support.esri.com/en/

ArcGIS Support: http://support.esri.com/en/

ArcDesktop help: http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html

Product documentation: http://resources.arcgis.com/content/product-documentation

Technical articles: http://resources.arcgis.com/content/kbase

About ArcGIS for Desktop: http://www.esri.com/software/arcgis/arcgis-for-desktop/index.html

An introduction to commonly used GIS tools:

http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_introduction_to_the_commonly_used_GIS_tools/002s00000006000000/

4.2 CARIS HIPS (Hydrographic Information Processing System)

Contributed by Cmdr. Izabel Jeck, Directorate of Hydrography and Navigation (DHN), Center of Hydrography (CHM), Brazilian Navy, Brazil

4.2.1 Introduction

The purpose of this document is to present to the user the basic workflow of the program CARIS HIPS (*Hydrographic Information Processing System*), version 7.0, to process multibeam data.

HIPS is a suite of comprehensive hydrographic data processing tools that can be used to process simultaneously multibeam, backscatter, side scan sonar, LiDAR and single beam data. CARIS HIPS is a program developed by a Canadian company named CARIS (www.caris.com).

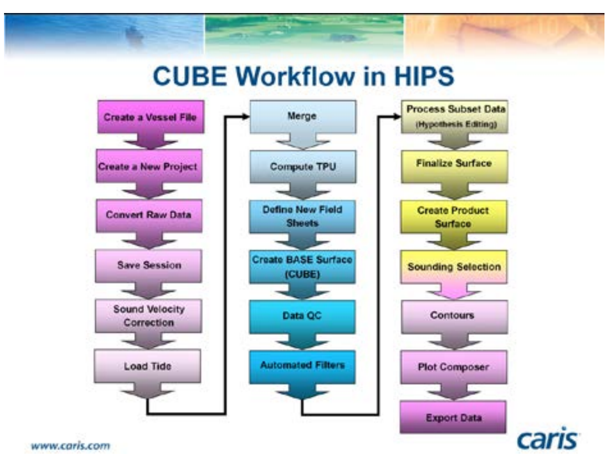
To use this software it should be note the computer minimum configuration, this information and any extra information can be obtained in the software documentation or in CARIS web page.

4.2.2 Acknowledgements

This document was made with the contribution and the authorization of CARIS - Universal System Ltd. Company, and further information can be obtained in www.caris.com.

4.2.3 HIPS Workflow

The figure below presents the software workflow.



Create a Vessel File (HVF): Setup the sensor locations and uncertainties in the vessel reference frame.
Create a New Project: Setup the *Project - Vessel - Day* data structure.
Convert Raw Data: Raw data is converted into HIPS data format.

Save Session: Save the current workspace (data and current view).
Sound Velocity Correction: Load and edit sound velocity profiles and apply the correction.
Load Tide: Load tide data from one or more tide stations.
Merge: Combine vertical and horizontal information to produce geo-referenced data.
Compute TPU: Utilize uncertainty values entered in the HVF in an effort to compute the total propagated uncertainty of each individual sounding.
Define New Field Sheets: Define the map projection and location of the data.
Generate BASE Surface (CUBE): Merged data is used to produce CUBE surface.
Data QC: Sensors, such as navigation, gyro, heave, etc. are examined if problems have been identified in the BASE surface.
Automated Filters: Filter soundings using swath geometry and/or according to IHO survey order accuracies.
Process Subset Data: Validate CUBE surface and edit the geo-referenced soundings directly, on many lines simultaneously, where the CUBE surface has been adversely affected by erroneous soundings.
Recompute: Update the BASE Surface after the data has been edited and the surface has been validated.
Finalize: Finalize BASE Surface to ensure designated soundings are carried through to bathymetric products.
Create Product Surface: Produce a generalized product surface from the BASE Surface.
Contours: Use either the BASE Surface, product surface or tile set to output contours.
Sounding Selection: Use a height source for selection of a representative sounding set.
Export Data: HIPS soundings and surfaces can be exported to various formats for data transfer.

Before start the processing the user must know:

- dX,dY,dZ sensor offsets
- echo sounder used and number of beams
- sensor timing corrections
- sensor bias corrections
- positioning format (geographic or ground coordinates recorded?)
- have calibration offsets been applied?
- has vessel motion been removed?
- has remote heave been removed?
- has draft been applied?
- has sound velocity corrections been applied?

4.2.4 Project Data Tree and General Information

The Control Window displays the data in an expandable tree. This display is organized to reflect the Project, Vessel, Day, Line directory structure. Data deleted from the Project tree is sent to the Recycler folder as defined in an Environment variable in the Options dialog.
Project information can be viewed in the **Project Properties** dialog box. General project information and information about the coordinate system is stored here. The project geodetics can be transformed using this dialog box if required.

Default Options

- Presentation and Performance

The default options can be accessed using the **Tools > Options** menu. The **Options** dialog box is organized into six tabs.

The **General** tab controls zoom and pan factors, and settings for displaying sounding size, the SVP location symbol and the navigation point symbol. The pick aperture setting determines how close the cursor has to be to an item in order to select it. A number of other options can be enabled or disabled in the General tab.

The **Display** tab controls the colour of lines and backgrounds in different parts of HIPS and related editors.

Under the **Display** tab the user can configure the presentation of Geographic or Projected Grids & Scale Bars for the HIPS Display window. Grids, which can be displayed as lines, ticks or crosses, will be displayed, with a scale bar, appropriately for the current zoom factor. Under the **Units** section users can configure the display units for their data.

All data in HIPS is stored in metres and so the conversion will be applied to the display of the data.

- Directories and Environment

The **Directories** tab enables the user to modify the location of project dependent data files, either one at a time or by using the **Root Path,** to set all the HIPS and SIPS data paths at once, if they all have the same base path. This information is saved to the registry.

The **Environment** tab enables the user to modify the locations of the HIPS and SIPS support information, such as the master file, symbol file, datum file, scratch location, etc.

By default, this information is located in the ..\CAIRS\HIPS\System directory.

- S-52 and S-57

The **S-52** tab controls the display of S-57 ENC data, there are options to select colours and transparency for depth areas, symbols, contours etc.

The **S-57 Environment** tab controls the location of S-57 dependent configuration files.

– 3D View

The **3D** tab controls the behavior of the keyboard and mouse controllers when navigating the 3D View:

• **Controller Type**

- Terrain Flyer: navigate from the perspective of the height source.
- First Person: navigate from a camera view of the current 3D scene.

• **Flight Speed:** Controls the speed at which you move through the display during a fly-through.

• **Zoom Speed:** Controls how fast the view is changed when using the zoom controls.

• **Smooth-Fly-To Speed:** Controls how fast the view is changed when a user double-clicks an area of the 3D View.

Open Background Information

Both vector and raster information can be loaded into HIPS as background information. This option can be accessed using the **File > Open Background Data.**

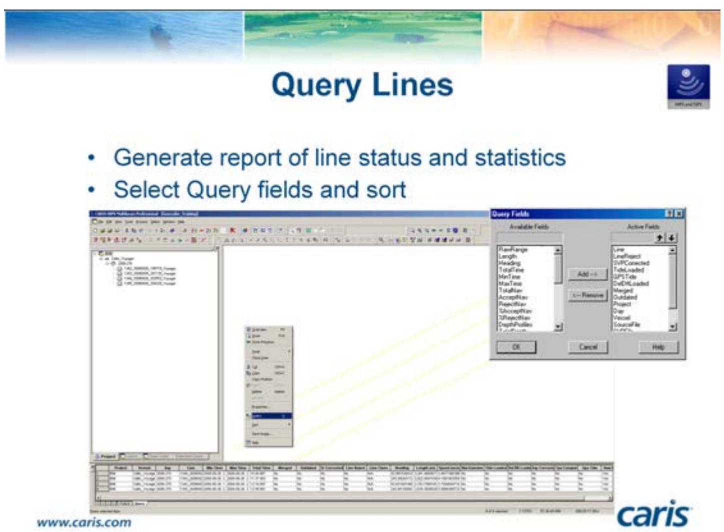
Layers Data Tree

The **Control Panel** displays the data in an expandable tree. This display is organized to reflect the layers of data displayed in the HIPS Display Window. It is possible to toggle these layers on and off using the check boxes in the **Layers** tab, and the draw order of the layers can be changed by dragging and dropping the layers in the **Draw Order** tab.
Entire Fieldsheets can be turned off from the right-click context menu.
In order to select data such as track lines in the Display Window**,** the layer must first be highlighted in the **Layers** tab of the Control Window.

Selecting Data in HIPS and SIPS

Survey lines can be selected by highlighting them in the Project tab of the Control Window, or by clicking on them in the Display window. Multiple lines can be selected using the shift and control keys. Lines can also be selected by dragging a box over the lines in the Display Window.
Selection in the Display Window can be done in two ways: **Select by Range** or **Select by Lasso.**
Lines cannot be selected when the cursor is in constant zoom mode. When in constant zoom mode. To exit from zoom mode, click on the zoom icon or use the ESC key.
Other selection options are **All in Display, All and Clear Selection.**
Note: The Select by Range tool is depressed by default. If you are unable to select lines in the Display Window check to see that this tool is selected.

Query Lines



All the information listed when a line is queried is calculated on-the-fly so it may take a few minutes to list. The data is displayed in the Worksheet Window and the columns can be sorted by clicking on any of the headings. By right-clicking on the headings and selecting **Query Fields…** from the pop-up menu you can choose which fields are displayed. The options **Select All**, **Copy** and **Paste** and **Save As…**allow the contents of the Worksheet Window to be taken into software like Excel.
Right-click in the Display Window brings up a pop-up menu you can use to copy the geographic coordinates of the cursor position to the clipboard.

4.2.5 HIPS Workflow- Detailed

Vessel editor

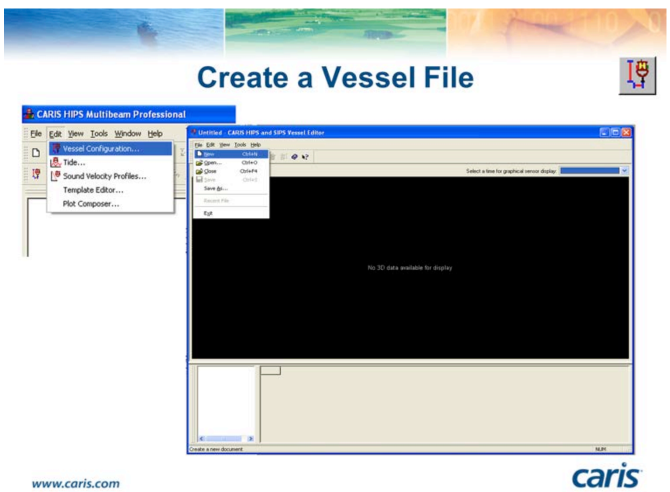
The Vessel Editor is used to create and edit HIPS Vessel Files (HVF). The HVF contains information necessary for combining all sensor data to create a final position/depth record. Sensor information is used during the merge process. All sensor entries are time stamped, therefore one HIPS vessel file can be used for the life of a vessel. If a sensor is moved or added, the new offset information is given a new timestamp. The merge process compares the date and time of the observed data with the date and time of the HVF sensor information to ensure that the appropriate offsets for that time are used.

Note:

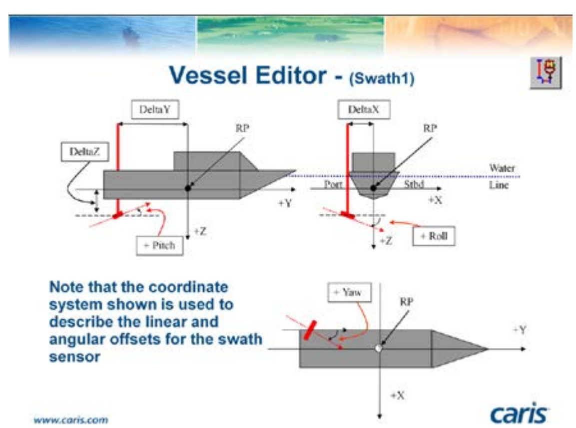
It is not possible to process sensor data time stamped prior to the earliest HVF timestamp, as the program applies the most recent information.

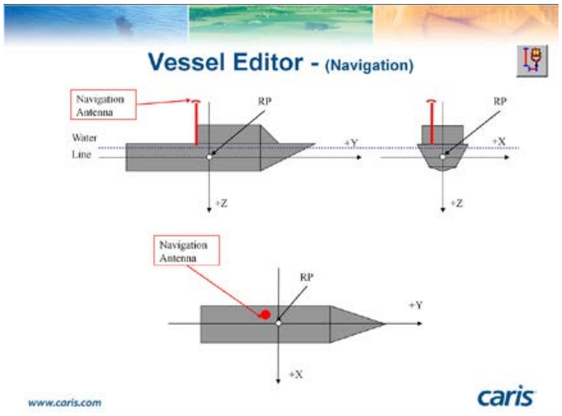
The positions of the sensors in the HVF may not be the same as the actual configuration of the vessel. For example, if sensors offsets are applied in the acquisition software, the HVF should describe the 'corrected' position, not the actual position of the sensor.

The vessel editor can be accessed using the edit > vessel configuration…from the main menu.

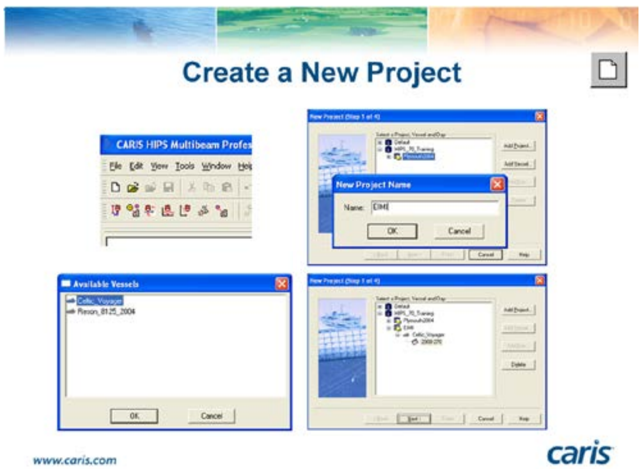


The figures below present HIPS coordinate system to be considered in the VESSEL EDITOR - SENSOR SECTIONS)



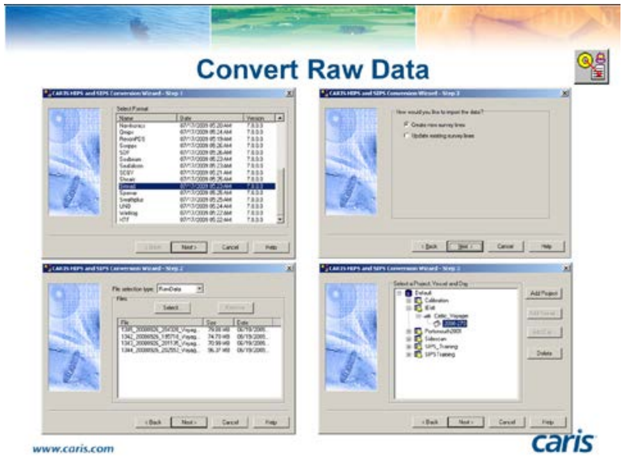


Create a New Project



A list of existing projects will be displayed in the option **file > open project**, you can also create a new project using **file > new.**
Note: Do not use spaces in the name of the project.
In the **file > new,** highlight the project created, click the **Add Vessel…** button and select the hvf vessel file; click the **Add Day…** button. Select the geographic parameters for the project.
Note: Regions of the world appear in the **Group** list. When a group is selected, the user can select from a pre-built set of map **Zones**. These parameters are referenced by the Map Definition File (Hips\System\mapdef.dat). This file can be appended to if additional definitions are required or another file can be selected by accessing **Tool > Options** from the main HIPS menu and then selecting the **Environment** tab.
The user may also opt to have the program automatically select the appropriate UTM zone.
Define a geographic window for the project. The default region is the entire globe. If an existing project in the area is open, the **Current View** button can be used to get the extents.

Convert Raw Data



- Select the **Conversion Wizard** icon or **File > Import > Conversion Wizard...** from the menu to start the conversion wizard.
- Select **data format** from the list of formats.
- Select the **File selection type** from the list, in this case **RawData**
- Select all the lines.

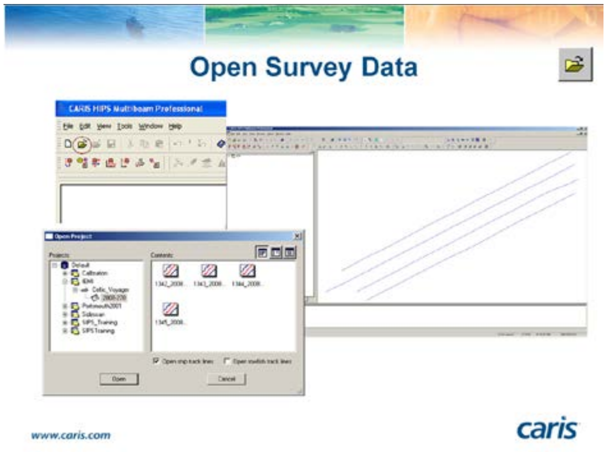
Raw data files are usually stored in PreProcess so this is the default location.

- Select the destination Project, Vessel, and Day folder for the converted
- Select the geographic reference system used for the acquisition of the raw data. If Ground Coordinates are used, the appropriate zone definition needs selecting from the list. If Geographic Coordinates are used nothing more needs to be set.
- Do not set **Navigation** or **Depth** filters. The **Navigation** filter is useful to restrict data to a specific geographic region. The geographic region can be defined manually or by Project File extents. The **Depth** filter is used to reject large bathymetric spikes that fall outside of the min and max depths of the survey area.

The first five steps of the Conversion Wizard are the same for all data formats. From Step 6 the wizard differs to accommodate individual requirements for each format type.

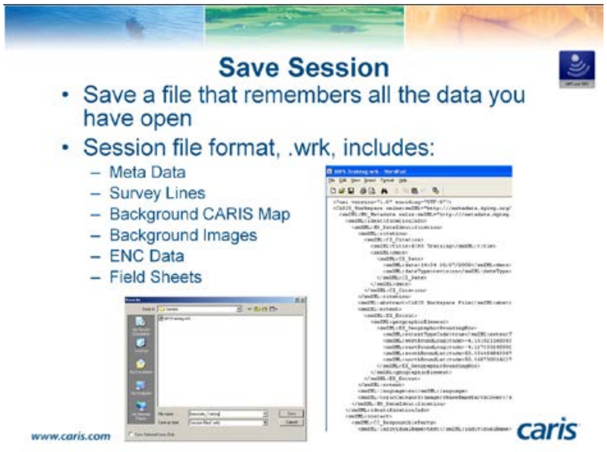
The **<Back** button can be used to check that no mistakes have been made in the settings. Once the line data has finished converting the **<Back** button changes to **<< Restart**.

<u>Open Survey Data</u>



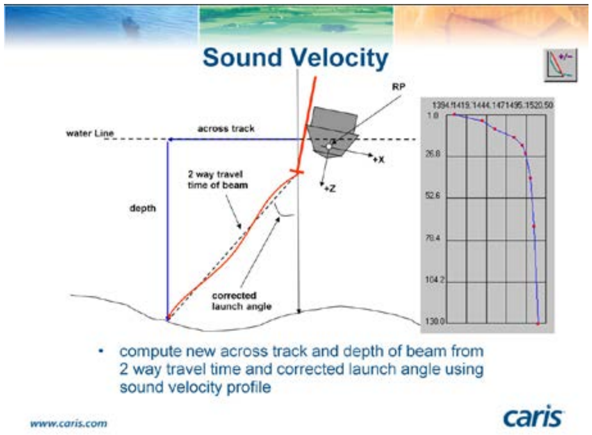
Once converted the data can be opened in the main interface.
The lines will be listed in the Contents window.
If side scan data was converted and resides in the same files as the bathymetry data then the tracklines showing the vessel and towfish positions should both be selected.

Save Session



The session file is a text file located in the HIPS\Session directory. It consists of a series of data objects in XML format describing:
• **Workspace –** Creator of the session, HIPS project name, creation and last modification times, and location of the HIPS source data.
• **View Extent –** The last used window size and location.
• **Project Data –** Survey lines, CARIS files, background images, and field sheets.
Double clicking a session file in Windows will automatically open the session in HIPS.
Sessions should be used to efficiently organize the data that you are working with.
Having a series of small work areas as separate sessions instead of one large project will allow for faster, more effective data access.

Sound Velocity



Raw data formats such as Simrad contain two-way travel time and beam launch angles.
Producing a geographically referenced sounding position and depth from this data is a two-stage process.

• The procedure for calculating the length and path of the sound wave through the water column for each beam is called the Sound Velocity Correction (SVC).

The result is an along-track/across-track/depth for each beam.

• The Merge process (which is explained later) converts the along-track/across-track/depths into latitude, longitude, depth by combining the ship navigation with horizontal and vertical offsets in the HIPS vessel file.

The SVC algorithm calculates the ray path of the sound wave through the water column for each beam. In order to do this the program needs to know several things:

• Any rotations that have to be applied to the sensor head – both static (i.e., setup and calibration parameters) and dynamic (measured by the attitude sensors).

• The acoustic velocity of the water column. This information is loaded into the program as a sound velocity profile.

There are different options for applying velocity information to a swath:

• **Previous in time –** This is the method traditionally used by HIPS, which applies the profile taken immediately before the collected swath.

• **Nearest in time –** In this situation HIPS will apply the SVP with the time stamp closest to that of the collected swath.

• **Nearest in distance –** Uses the position attributes of the profile to determine the nearest profile to a given swath.

• **Nearest in distance within time –** Uses the position attributes of the profile to determine the nearest profile to a given swath within a time window which the operator has to enter.

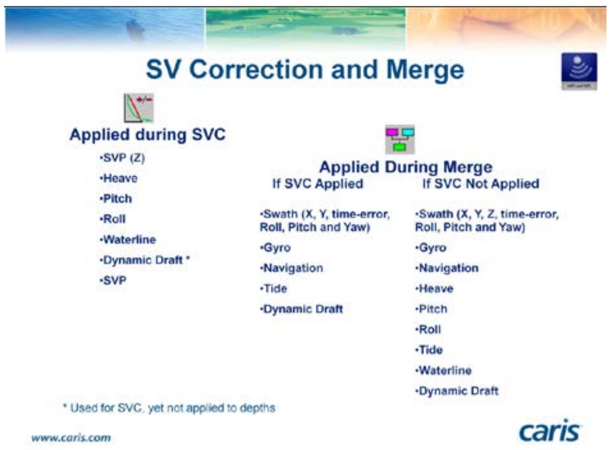
Note:

The profile selection methods listed above are made on a swath-by-swath basis and so it is possible to have several SVPs used in the same survey line.

For Simrad data, the sound velocity and attitude corrections have been applied during acquisition, and do not need to be reapplied.

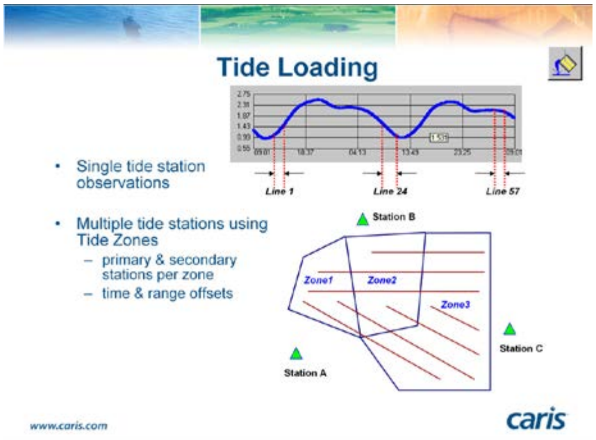
The process checks the HVF to see if the heave, pitch, and roll are set to Apply “Yes”. Since Simrad has already corrected for Sound Velocity, it does not have to be re applied. Doing so would skew the data.
Note that in this case there is no sensor called SVC to the Vessel File (hvf). This sensor would need to be configured in order to apply (or reapply) sound velocity.

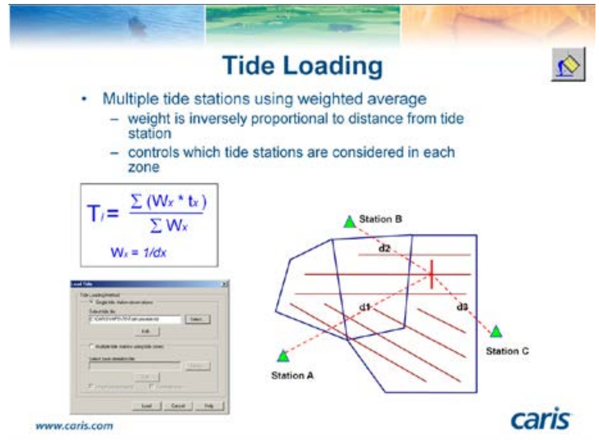
Sound Velocity Correction



As stated earlier, some of the parameters that are usually applied in the merge process can be applied during the SVC process.
If you sound velocity correct your raw data to produce a new observed depth, the merge process knows what has been applied and will not reapply it as SVC is optional.
The following are taken into account during SV correction:
• Travel Time - angle observations
• Sound Velocity Profile
• All entries in the SVP section.
• Water line
• Dynamic heave, pitch and roll, along with any errors. Including pitch and roll induced heave calculated from MRU lever arm offsets (X, Y and Z of motion sensor)
• Dynamic Draft - uses a table of speed versus draft values in the HVF.
• The Dynamic Draft values are only used to find the proper sound velocity layer to start SVC. The draft values will be applied to the observed depths in the Merge Process.
Note: The dynamic heave, pitch and roll will only be applied if **Apply** is set to **Yes** in the HVF for each individual parameter. However, the offset values for Roll, Pitch and Yaw entered into the Swath/SVP will always be applied.

Tide Loading



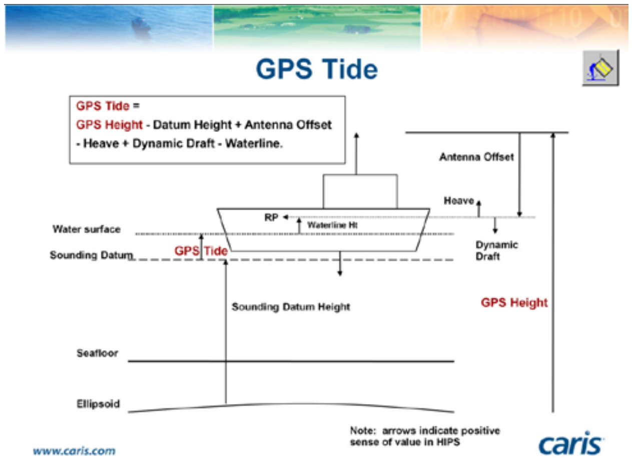


Tide data can be applied in two ways:
• The tide data from a single tide station can be applied to each survey line as a whole.
• Tide data from multiple tide stations can be compiled. The tide data is loaded for each part of a line as determined by the zone the part falls into.
To define a tide zone several parameters are required:
• Zone boundary in geographic coordinates.
• The location of the primary tide station.
• Locations of up to 3 optional secondary tide stations.
• Time offset in minutes for each station.
• Range offset / tide scalar for each station.
• Outage limit in minutes controls when data will be extracted from secondary stations.
• Interpolation interval in seconds for final interpolated tide loaded into each survey line data structure.
Note: When using data from multiple stations, a weighted average of tidal observations along the survey line is generated. This only works if the information about the tide coefficients is given in the tide zone file. The weight given to the tide data is inversely proportional to the distance between the station and the swath data, which is why the zone definition file should include latitude and longitude positions for tide stations.

To apply tide you have to select all of the lines and click on the **Load Tide** tool button or select **Process > Load Tide**.

The tide weighting formula can be expressed as the following;
Ti = The tide value for the ith profile along the line
tx = The tide value from the xth tide station for the time and location of the ith profile
Wx = The weight of the tide value coming from the xth tide station
dx = The distance between the ith profile along the line and the xth tide station.

GPS Tide



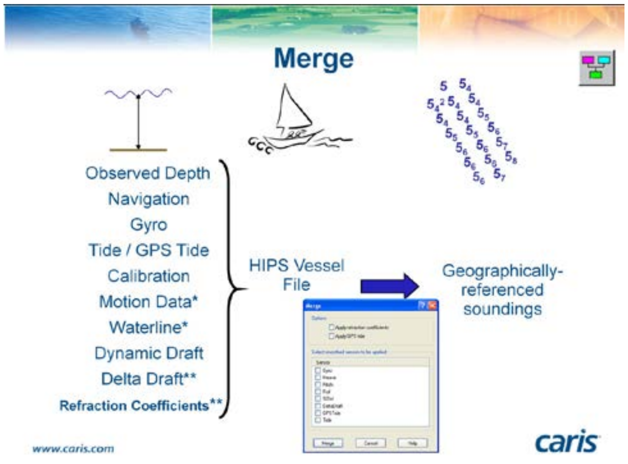
HIPS has the ability to use ellipsoid heights from GPS RTK instead of traditional tide data.
GPS Tide data is applied optionally as part of the Merge process. It is used as a direct replacement for regular tide data. The vessel configuration is used as normal with respect to transducer offsets, calibration values, application of heave, pitch, and roll, etc.
In order to apply GPS tide you first need to calculate it. This calculation is performed with the **Process > Compute GPS Tide…** command.
To be able to use the GPS height information for tidal corrections, the separation between the ellipsoid and the vertical survey datum needs to be known. When computing the GPS tide this separation can be entered as a single value or a model file can be used. Two binary sounding datum models can be used for computation of GPS tide:

• GEOID99
• EGG97 (European Gravimetric Geoid 1997)

Alternatively, an ASCII sounding datum model can be used. The data must be in the format Lat, Long, Z, for example: -33.848326,151.192435,20.4
Not in all cases raw GPS height will be recorded in the raw data files. The used acquisition system may already have made corrections on this data. To allow for this it, is possible to select which terms of the GPS tide computation should be applied and which not. If a sensor has already been applied on the GPS heights by the acquisition system, it should not be applied by HIPS. Only corrections that are selected in the Compute GPS Tide dialog will be applied in the computation.

Merge

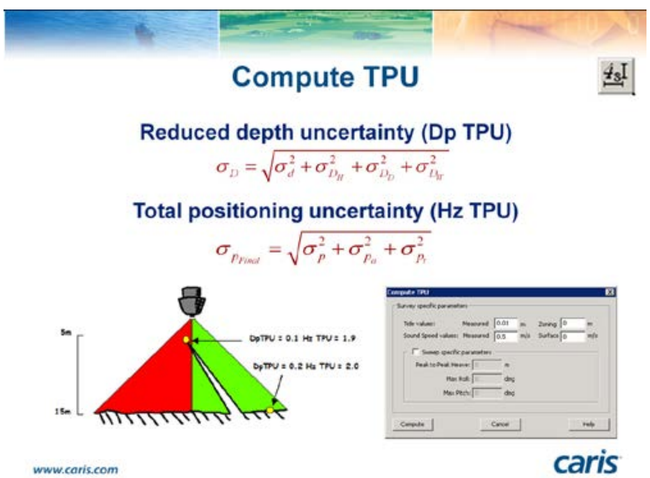
Merge takes into consideration all of the horizontal and vertical offsets in the HVF to produce a processed sounding.



Parameters marked with an * have been applied in the SVC process. Some additional parameters need to be selected as part of the Merge process. These include:

- Apply Refraction coefficients
- Apply GPS tide
- Smooth Sensor data

Compute TPU



In order to generate TPU values for each sounding, the definition of uncertainty estimates for each of the contributing sensor measurements has to be combined using a propagation algorithm. The result is a separate uncertainty estimate for the depth, **DpTPU**, and horizontal position of the sounding, **HzTPU**. The results are presented at the 95% confidence interval which is equivalent to 1.96 x the standard deviation.

The compute TPU function must be performed after the Sound Velocity correction. Whether it is applied before or after Merge will depend on your adopted workflow. When you compute TPU it will always be done on the Observed depths, therefore tide and draught offsets will not affect the values.

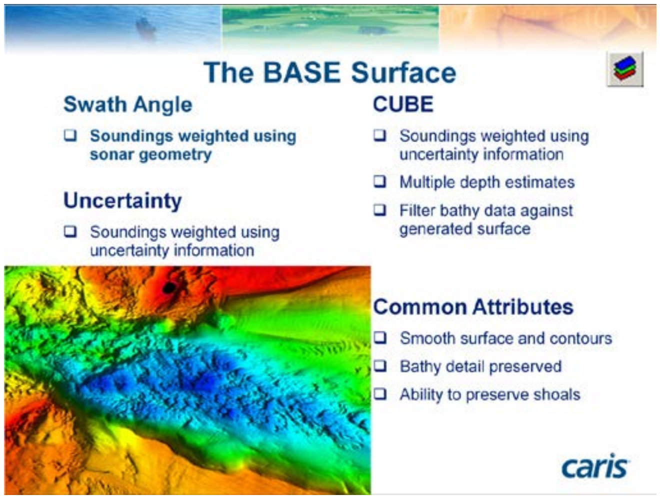
This calculation is performed with the **Compute TPU** tool button or **Process > Compute TPU**. The TPU values will be calculated for the soundings based on the errors defined in the HVF, the Total Propagation of Uncertainty algorithms and the survey specific uncertainty estimates defined.

Defining New Field Sheets

Field sheets are essential for the creation of Weighted Grids and BASE Surfaces.
Field Sheets can be created in locations other than the directory directly under HIPS.
To create a New Field Sheet the user can use the **New Field Sheet** tool button or select **Process > New Field Sheet.**
- Enter a Name for the Field Sheet.
Note: Field Sheet names should be made up of alphanumeric characters only and should not contain spaces.
- Enter a Scale.
- Leave the default values for the **Horizontal** and **Depth Resolutions**. This will define the measurable accuracy of the coordinate and depth resolutions.
- Confirm the **UTM Zone / Mercator**
- Define Field Sheet extents by three ways:
1) Select the Current Display button to use the current display as the perimeter.
2) Define the extents with the mouse by clicking on the User Defined Area button.
3) Type in the coordinates for the bottom left and top right corners of the field sheet.
- Create a user defined Field Sheet large enough to accept all of the data in the survey area.

Once a new Field Sheet has been created it will appear as a new layer in the Layer tab of the Control Window. The display of the Field Sheet name and outline can be controlled by the Properties dialog box.
The Field Sheet icon in the Control Window is green, meaning that this is the active Field Sheet. If you have more than one Field Sheet open then it is possible to change which is set as active. This is achieved by right-clicking on the Field Sheet name in the Control Window and selecting the **Set as Active Field Sheet** option from the pop-up menu.

The BASE Surface



HIPS supports the ability to create three different surface types:

- the Swath angle surface,
- the Uncertainty surface, and
- the CUBE surface.

Through the BASE Surface Wizard the user can select which type will be created. The surface types use different weighting algorithms to produce gridded images of the sonar data. All three surface types will produce a smooth surface that retains the sonar resolution. The user also has the ability to preserve shoal or critical values.

The CUBE surface is also a powerful, semi-automated cleaning tool that can be used to increase processing efficiency. Both Swath and Uncertainty surfaces can also be used for cleaning to some degree.

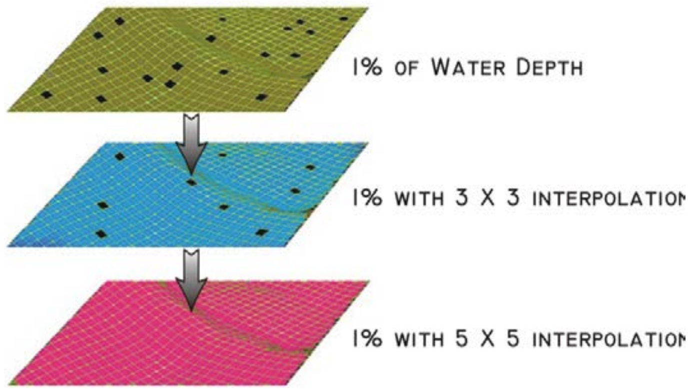
BASE Surface

When the cursor is moved over the BASE Surface, tool tips are displayed showing the value for the nearest Base Surface node. This option can be toggled on/off in the **Tools** > **Options** menu option under the **General** tab.

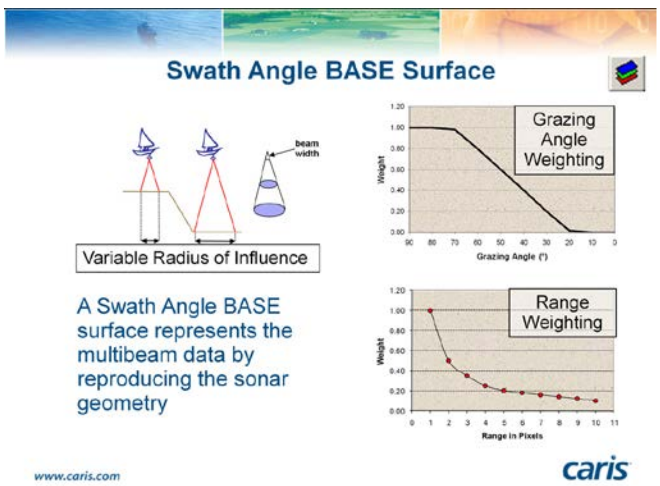
BASE Surface nodes can also be selected and queried. To query surface node values, enable the display of the surface in the Control Window and highlight that layer. Left click and drag to select an area of nodes. Select the **Edit** > **Query** menu option to have the XYZ values of the selected nodes displayed in the Query window.

Note: To obtain the maximum BASE Surface resolution it is often best to create a small BASE Surface and work upwards. If working with a single resolution the first BASE Surface you create may be 1% of the average water depth. The other option is to create a surface with a resolution that varies with the water depths of the survey area.

The initial BASE Surface may contain holes; the holes represent areas where the data was not dense enough for the resolution selected. In such areas you can interpolate to fill in the holes.



Swath Angle Surface – Theory



The Swath Angle BASE surface is a rasterization method specifically designed for multibeam data. It is an accurate representation of multibeam data because it considers the actual geometry of the sonar system. It does this in three ways:

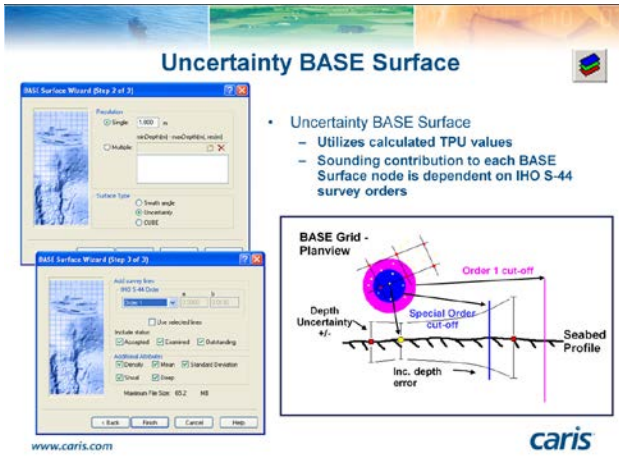
Variable radius of influence - The radius of influence of each sounding is calculated from the beam width of the selected sonar, and increases with depth and grazing angle. This addresses the increase in footprint size with distance from the sonar head.

Range weighting - The range weighting for each sounding decreases with distance from a node. This lessens the effect that soundings further away from the node will have on the BASE surface.

Grazing angle weighting - Errors in multibeam data tend to increase in magnitude in the outer beams due to the longer ray path distances. This tends to magnify refraction problems and other errors. The grazing angle weight function is applied to each sounding to reduce the effects of the outer beam soundings on the BASE Surface.

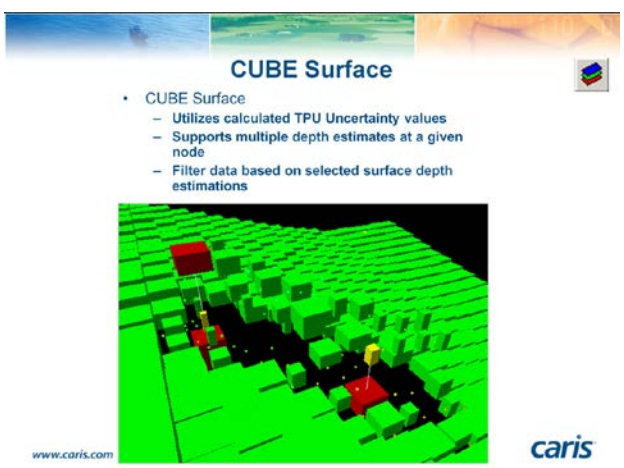
This weight function is controlled in an ASCII file called GrazingAngleWeights.txt in the HIPS\System directory. This file can be modified by the user

Uncertainty Surface – Theory



As mentioned previously, the horizontal TPU (HzTPU) and depth TPU (DpTPU) values are used in the creation of an Uncertainty BASE Surface. When creating an Uncertainty BASE Surface each sounding value will have an associated depth uncertainty and horizontal uncertainty. The depth uncertainty that each sounding has increases with the horizontal distance from the surface nodes. Therefore the greater the distance, the greater the uncertainty, and the less the sounding will contribute to the node.
The point at which a sounding will no longer contribute to a node is dependent upon the IHO S-44 Order selected. A sounding will only contribute to a node if the propagated vertical uncertainty of the sounding does not exceed the vertical accuracy requirement specified by the order.

CUBE Surface – Theory



The CUBE (Combined Uncertainty and Bathymetry Estimator) Surface can be used as a cleaning tool in the HIPS workflow, besides being an end product or for product creation.

During the CUBE Surface creation, soundings are weighted and contribute to surface grid nodes based on TPU values and distance from the nodes. The weighting method is similar to the Uncertainty BASE Surface creation.

The CUBE Surface however, allows for multiple depth estimates or hypotheses to exist at a single grid node, depending on the variation of the sounding data. CUBE then uses "Disambiguation" to determine which hypothesis at each node is the most "correct".

Users can verify, and if necessary override, CUBE decisions in Subset Editor. Once any necessary edits have been made to the CUBE surface, the CUBE filter can be applied to the data. Any sounding data that is not in agreement with the selected hypotheses will be flagged as rejected.

Before generating a CUBE Surface, all lines to be included must be merged and TPU must be computed.

- Highlight the field sheet name and then select the **New BASE Surface** tool or select **Process > BASE Surface > New.**
- Enter a name for the BASE Surface.

Note: It is a good idea to give the grid a name that relates to the resolution. Do not use spaces or non-alphanumeric characters.

- Enter a **Resolution.**
- Select **CUBE** for the **Surface Type.**

The Depth Filter can be used to limit the depth range of the generated surface to a specific interval.

- Set **IHO S-44 Order.**

- Select **Density & Locale** as the **Disambiguation** method.

- Click the **Advanced** button and ensure that the **Default Configuration** is selected on the Advanced Options dialog.
- Ensure that the **Initialization Surface** option is deselected.

The Disambiguation method will determine the method CUBE will use to determine the most likely bathymetry estimates. In this case it will use Density but if it is not confident in a selection it will utilize the Locale method.

The Advanced Options tab allows users to select or define the parameters to be used for the CUBE Surface creation. There are several pre-configured parameter sets available for selection. The configuration selection should be made based on the water depths and seafloor complexity of the survey area.

Note: Only experienced users should adjust or define the CUBE parameters. It is also possible for CUBE can use an Initialization Surface to ensure that grossly erroneous soundings will not be utilized in the CUBE surface creation. The CUBE surface will be visible in the Display window and the surface child layers, which are created by default, will be added to the Layers tab.

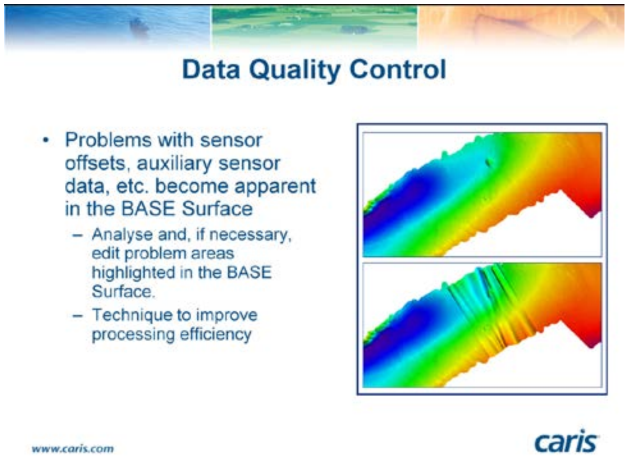
The CUBE surface contains the following layers:

• **Density:** The number of soundings that contributed to the selected hypothesis.
• **Depth:** The depth estimation of the selected hypothesis.
• **Hypothesis Count:** The number of different depth estimations at the surface node.
• **Hypothesis Strength:** A value to describe how mathematically confident CUBE is that the correct hypothesis was selected during disambiguation. Like Uncertainty, lower values are better.
• **Mean:** The mean of the soundings that contributed to the selected hypothesis.

• **Node Standard Deviation:** The combined standard deviation of the soundings that contributed to the selected CUBE hypothesis.
• **Standard Deviation:** The average standard deviation of the soundings that contributed to the selected CUBE hypothesis.
• **Uncertainty:** The vertical uncertainty associated with the selected hypothesis. Lower uncertainties are better.
• **User Nominated:** This layer indicates if the hydrographer has nominated a hypothesis during hypothesis editing. This process will be explained further in a later exercise.
• **Guide_Depth:** This layer represents the depth values of the initialization surface but resampled and displayed at the same resolution as the CUBE surface. This layer is only displayed if an initialization surface was selected during the surface creation.
• **Guide_Uncertainty:** This layer represents the uncertainty values of the initialization surface but resampled and displayed at the same resolution as the CUBE surface. This layer is only displayed if an initialization surface was selected during the surface creation.
• **Bounding Polygon:** This layer is a coverage polygon representing the boundaries of the data. The polygon includes the outer boundary of the data as well as an holes in the data.

The CUBE Surface can be shaded to assist in feature recognition. This and other image manipulation tools can be accessed via the Properties tab. This window can be used to manipulate the display of the various layers displayed in the Layers tab well as change the Disambiguation method used in the CUBE surface Creation.
Furthermore, the layer selected in the Layers tab will be displayed in the **Properties** tab. The following can be accessed when a BASE surface layer is selected:
• **Colour Map** and **Colour Range –** Control the display of images using **Colour Maps** (colours defined by pixel values) or **Colour Ranges** (colours defined by surface values). Colour Maps and Colour Ranges can be edited by click the Browse button to launch their respective editors**.**
• **Transparency (%) –** Set a transparency percentage to be applied to the display of the image. This is useful when displaying multiple images in the HIPS Display.
• **Depth Filter –** Filter the display of surface nodes by specifying a minimum and maximum depth value.
• **Raster Legend –** Enable the display of a Raster Legend to show the correlation between the surface values and selected colours in the HIPS Display window.
• **Shading and Vertical Exaggeration –** Shade the image to bring out surface texture. The shading can be changed by moving the sun symbol or entering values for Azimuth and Angle. Use the **Vertical Exaggeration** option to control the apparent vertical height of the surface, as this can also help to distinguish features.
Whether a surface is updated when changes are made in Subset Editor is defined using the “Keep Up To Date” flag for a surface.

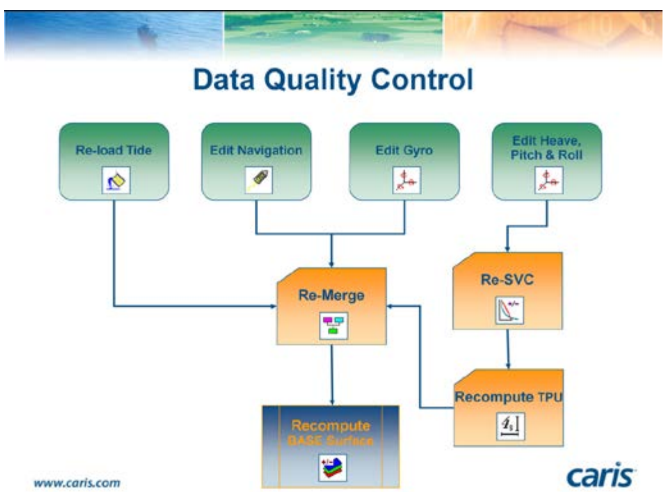
Data Quality Control



Prior to surface creation the auxiliary sensor data has not been investigated. The BASE surface can be used to highlight problems that may exist in the attitude and/or navigation data, problems which can be addressed using the Navigation and Attitude Editors.

By relying on the gridded data to highlight problems in the auxiliary sensors, processing efficiency can be improved, as the user is no longer required to investigate the motion and navigation of each survey line.

Note: A survey line must be selected before any of the line editors can be opened.



If changes are made to the navigation and/or motion data through the QC process it will be necessary to repeat previous steps in the HIPS workflow. The workflow diagram displayed here outlines the operations that need to be recalculated if changes are made.

Navigation Editor

The Navigation Editor lets you examine and clean individual position fixes as recorded by the vessel's positioning system.

- Select the first survey line in either the Control or Display Window.
- Select **Tools > Navigation Editor** or click on the Navigation Editor button in the Tools toolbar.

The **Navigation Editor** tab will be added to the Control window. This tab has four areas containing navigation editor controls:

• The **General** section allows the size of the navigation fixes to be changed. You can also specify if you want the points to be connected. You can select to display the graphs vertically or horizontally, and choose whether the minimum and maximum values displayed are for the entire line or the current view.
• The **Interpolation method** section determines the interpolation method used. This can be either linear or a tight, medium, or loose bezier curve. These are explained on the next slide.
• The **Spike detection** sections contain controls for finding speed jumps or time jumps.
• The final section deals with the **Number of points that are displayed** in the Display Window, this can either be based on a zoom factor or a fixed value. Users may also enable the display of Gyro Indicators and set the length and colour to provide an visual indication for the direction that the survey line was run.
When checking navigation, keyboard shortcuts can be used to speed up the procedure.

Navigation Interpolation:
Position observations do not usually occur at exactly the same instant in time as a depth observation (ping). It is also unlikely that there is a position for every ping.
In most cases it will be necessary to interpolate positions to match the times for each ping. The interpolation method is set in the Interpolation method section of the Navigation Editor tab.
The default method is **Linear**, where interpolation between each successive position is obtained by simply connecting the positions with a straight line. Linear interpolation is suitable when the original navigation positions are clean and do no significantly deviate from the neighboring positions.
The other three options (Loose, Medium, Tight) are various degrees of **Bezier Curve**. In these cases the line of interpolation will not follow the navigation positions exactly.

Edit Navigation:
There are four main edit functions that are used when editing data in the navigation or attitude editors.
• **Accept** - This will change the status of all selected sounding positions to Accepted. This can also be activated using the **A** key as a shortcut.
• **Reject - Break Interpolation** - This will reject selected sounding fixes without trying to interpolate their position. (Navigation for the pings is interpolated from the navigation data.) This function causes all soundings between the navigation fixes immediately before and after the gap to be rejected. Use **B** as a the shortcut key.
• **Reject -With Interpolation** - This rejects selected navigation fixes but interpolates a position over the gap. This can also be activated using the **W** key.
• **Query** - This gives the details of the fixes in the Worksheet window. The data can then be sorted in ascending or descending order by double-clicking the column heading. This can be useful for finding jumps and spikes. The **Q** key can also be used.
The whole line can be queried by right-clicking and selecting **Query Line** from the popup menu. This information is in memory, so it lists quickly, and the data can be pasted or saved as a text file for use in other software.

Attitude Editor
The attitude editor is used to display and edit sensor information.

You have to select a line, and then select the **Attitude Editor** tool button or **Tools > Attitude Editor.**
Gyro, Heave, Pitch and Roll sensor information is displayed by default. More sensors can be added by selecting them from the available sensors list, which can be accessed from the **Tools > Sensor Layout** menu.
Other sensor data could be GPS Tide, Side Scan Gyro, ROV Altitude etc.
The four default windows in the Attitude Editor display time series graphs for Gyro, Heave, Pitch and Roll. The tools to manipulate this data are available in the Control Window under the Attitude Editor tab.
When the cursor is placed on one of the time series graphs, the value is displayed as a tool tip.
Gyro data in CARIS HIPS is displayed as a positive value when a clockwise rotation is experienced.
Heave data in CARIS HIPS is displayed as a positive value when the vessel is heaved upwards.
Pitch data in CARIS HIPS is displayed as positive when the vessels bow is down. Roll data in CARIS HIPS is displayed as positive when the vessels starboard side is up.

Filtering:
The **Moving Average** option works by averaging data within a window. The window size can be based on seconds or number of data points. Increasing the size of the window increases the level of smoothing.
The **Fast Fourier** option takes out all frequencies which have a shorter wavelength than the box size set in points or seconds.
The **Threshold** option sets the cut-off value as a multiple of the standard deviation. The cut-off value is applied to the difference data. The **Difference** between the filter line and the sensor data is then calculated. This difference will be greater in areas in which local variability is high. Any difference values outside of the threshold limit will be rejected with interpolation.
Filtering can be applied in three ways: either the present screen only, filter to the end of the line, or filter the whole (selected) line. Filtering can be applied with or without interpolation. As in the Navigation Editor, sensor values may be **Queried**, **Accepted**, **Rejected** (with interpolation), or **Rejected** (break interpolation).
Motion data that contains high-variability noise can be addressed by using the smoothing routine. Smoothing parameters will be stored separately and applied during Sound Velocity Correction.

Swath Editor
The Swath editor is used to view and edit the multibeam swath data. This can be either a manual process or a semi-automated one using filters based on knowledge of the data.
After select the first survey line in the Project tab of the Control Window click on the **Swath Editor** button or select **Tools > Swath Editor.**
The Swath Editor has several windows:
• **Swath Editor** tab - A new tab appears in the Control Window, containing a number of sub-panels that are used to manipulate the data in the Swath Editor.
• **Plan View -** Displays the beams and swaths in plan view. The bottom of the display is the start of the line. The scroll bar can be used to page through the data.
• **Side View -** This is an along track profile of all the swaths in the Plan View viewed from the starboard side, so that forward is to the right of the window.

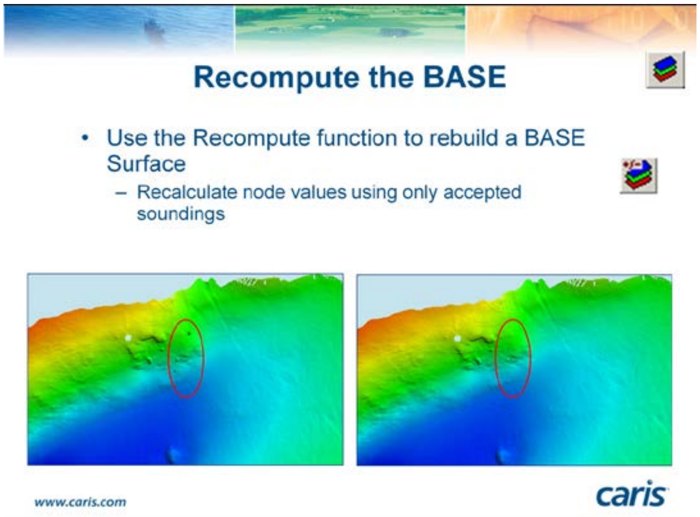
• **Profile View -** This view represents a single swath at a time viewed from the rear. The scroll bar can be used scroll through the individual profiles.
• **Rear View -** This is the data displayed in the Plan View window viewed from the rear of the ship.
• **Amplitude -** If the sonar data contains intensity information then it can be displayed in this window. This view can assist in feature recognition.
• **3D View** - In addition to image manipulation in the 3D View, the data can be selected, rejected, queried and accepted.

Subset Editor
The Subset Editor is activated by either clicking on the **Subset Editor** icon or by selecting **Tools > Subset Editor > Open** from the main menu.
A new Subset tab will appear in the Control window, and an additional toolbar will be displayed. The cursor in the Display Window will change to crosshairs to define the outline of the subset to be edited.
To define a subset drag a box around the data to be included using the left mouse button.
The subset outline can be rotated by holding down the **Ctrl** key while drawing. The subset can also be resized by dragging the corner anchor points.
The yellow 2D profile window can be resized or moved by dragging the anchor points.
To move the 2D slice along the subset use the keyboard arrows.
The toolbar has the following options:
• 2D and 3D windows can be toggled on and off.
• The status of subset tiles can be set to incomplete, partially complete or complete.
• The subset can be locked, so that the no accidental resizing, moving, and rotation can take place causing a redraw. The subset can also be locked by pressing the <**L**> key.
The Subset Editor tab has the various Subset Editor items organized into a **Data** tree. By selecting the items in the **Data** tree, the corresponding properties will be displayed in the **Properties** section of the Subset Editor tab.
Subsets should not be too large. If they contain too many soundings they will take too long to open.

Editing windows:
The user can pan (middle mouse button) and zoom in/out (Crtl+right mouse button) on the **2D View**. Other functionality includes:
• The **View Direction** of the data in the window is relative to the subset orientation. The yellow arrow on the subset outline in the Display points towards the top. The view direction should be selected with this in mind. Select **Automatic** to always have the 2D View display along the longest edge of the slice.
• Set the display by selecting **Auto** Exaggeration, **min/max** values or by adjusting it manually using the Vertical Exaggeration in the 2D View. You can also choose to **Include Rejected** soundings in the scaling. Soundings can be displayed as digits and graticule labels can be turned on and off.
The **3D View** is controlled in the same way as the 3D View for the Swath Editor. There are however functions that are specific to the Subset Editor.
• The **Data Display** options allow for the 3D view display **Type** to be set as Points, Spheres, Cylinders, Surface or Surface Wireframe. The Subset **Outline** and **2D view slice** can also be

enabled for display in the 3D View. **3D Dynamic Skip** reduces the number of points displayed while the 3D View is being manipulated by the user.
• Under the **Controls** the user can **Allow rotation under data.** This allows the operator to rotate the data beyond the horizontal and view the underside of the sounding data. The display of the Compass, Lighting and Exaggeration can also be enabled/disabled.

Recompute the BASE Surface



After you have finished editing the data in the Subset Editor and exited out, or cleaned the bathymetry using the Surface filter, the BASE Surface may appear outdated. This is indicated by a red exclamation mark on the BASE Surface icon in the Session tab of the Control Window. The **Recompute** function enables quick recomputing of a BASE Surface to update any changes that have been made.
If all manual edits had been performed in the Subset Edit with the **Automatic BASE surface update** option selected under **Tools** > **Options** the surface would already be up-to-date and would not need to be recomputed. **Automatic BASE surface update** recomputes small sections of the surface as the user moves across the point cloud in Subset Editor.
If the surface is outdated select the BASE Surface from the Control Window, right mouse click and select **Recompute** from the pop-up menu.
All changes made during the editing process will be applied to the BASE Surface.

The data should now be geo-referenced and cleaned.
The HIPS data can now be used to create bathymetric products. These are generated in the Field Sheet Editor.
Bathymetric products include Selected Soundings, Contours and others.

Product Surface

The first step to generating smoother contours is to create a Product Surface. The Product Surface maintains the designated soundings from the finalized BASE Surface and can be used to create more cartographically correct contours. A generalized surface is created by rolling an

imaginary ball over the BASE Surface. The user can specify the scale and radius of the ball. The ball will be rolled over the surface at the interval defined by the resolution.
The Product Surface is activated highlighting the Field Sheet layer and selecting **Process > BASE Surface > New Product Surface**.

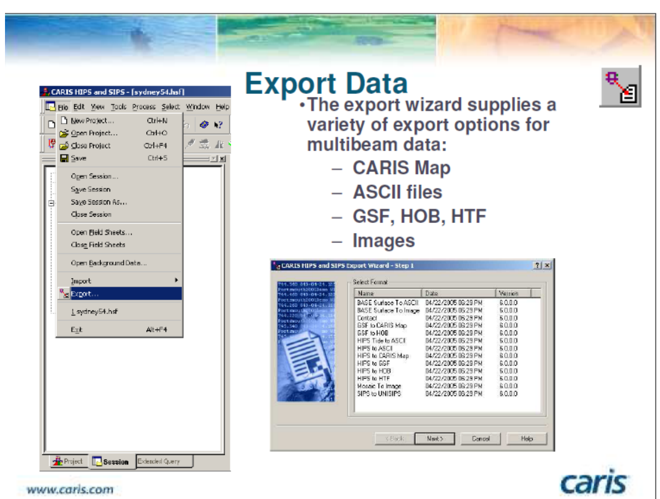
Sounding Selection

It is impossible to display all soundings produced by a multibeam survey on a map or chart and maintain legibility. The selection of soundings that appear should be dependent on chart scale and purpose. Soundings can be created from a Tile layer or BASE Surface.
Click on the **Create Sounding Layer** icon or select the **Process > Products > Soundings > New Sounding layer.**
Soundings can be displayed in two ways, the soundings can be engineering style, with decimals, or they can appear as slanted soundings with drop decimeters.
Note: In most situations it is wise to use the same data source for soundings and for contours. This removes the possibility of producing soundings on the wrong side of contours.

Contouring

Contours are created using a three step wizard.
Open the Contouring wizard by clicking on the New Contour Layer icon or by selecting the **Process > Products > Contours > New Contour Layer.**
Contouring from different sources produces different results. The BASE Surface (swath angle method) is a mean surface, whereas tiles or bins can be shoal-biased. If you are creating products that will be used for navigation then a bin or uncertainty BASE Surface maybe a more appropriate source for building contours.
- Select the Depth layer of the generalized Product Surface as the height source.
- From the **Standard Contour** tab select an interval.
- The **Theme Number** will determine how layers are named and organised, this information can be used by CARIS GIS to differentiate the data.

Export Data



This will launch the Export Wizard. There are various export options available. Data, grids, and images can all be exported using this wizard.
The BASE Surface generated in HIPS can be exported to image file or ASCII file. The user can specify the units for the exported depth values and additional sounding information can also be exported as optional attributes.
The depth can be exported as an ASCII file.

4.3 Generic Mapping Tools (GMT)

Generic Mapping Tools (GMT) (Wessel and Smith, 1998) is a collection of open source mathematical and mapping routines for use on gridded data sets, data series, and arbitrarily located data. The GMT package is available for download from the University of Hawaii website (http://gmt.soest.hawaii.edu/).

4.4 r2v, Surfer, Global Mapper

Contributed by John Hall, Geological Survey of Israel (Retired), Israel

Recent compilations of the Caspian Sea and the Red-Arabian Seas illustrate the preparation of high resolution grids using the r2v (www.ablesw.com), Global Mapper 12 (www.globalmapper.com), Surfer 9 (www.goldensoftware.com) software.

An overview of each of the packages follows:

4.4.1 Able Software Corporation's r2v (raster to vector)

A license for r2v costs $395 for an academic user, or $795 for non-academic. A USB dongle is now employed. As used for bathymetry, a scanned map is geo-referenced, either using Global Mapper (more convenient), or r2v. Once a .tfw (true world file) exists, r2v can work in map coordinates, which are usually Mercator. Very few of r2v's capabilities are used for this work. Mainly the automatic line following is used, following conversion to grayscale, and then contrast selection to accentuate contours while eliminating background. The program will rapidly follow a line until a gap or branch is encountered. One then toggles to manual mode, passes the obstruction, and continues in automatic mode. The program gives complete control over the nodes, adding, moving, smoothing, deleting, and joining. Use of the F2 and F3 keys allows almost unlimited magnification for very steep slopes. Navigation across the map is very easy. Contour values are easy to apply, either individually, or by identifying a beginning value and increment and then drawing a line across sequential contours. For complicated contours it is possible to color code the contours. Output of r2v is generally via a .dxf vector file, or as .xyz points, although other formats are available. The program is also able to digitize points, but each point requires a number of steps. Dr. Ted Wu is the owner of Able Software and readily answers questions. A trial version of r2v can be downloaded.

4.4.2 Global Mapper Software LLC - Global Mapper 12

A license for Global Mapper 12 is $349, and somewhat less expensive for multiple licenses. Global Mapper was started in 1991 by Mike Childs, who is very responsive to additions and modifications to the program, which undergoes regular improvements. In short, GM12 is a comparatively easy to use program which deals with a wide variety of raster images, grids and vectors, especially those available on the Internet. It deals with a continually increasing number of projections and datums, as well as formats, and allows speedy conversion between them. For our purposes, GM12 allows rapid geo-referencing of charts, rapid TIN display of .xyz files prepared by r2v, projection change of .dxf files, rapid display of multiple grids, superposition of various raster and vector layers, highspeed input of grids (ASTER-GDEM, SRTM, SRTM30-Plus, GEBCO_08, IBCAO etc.), and their manipulation (cropping, clipping, translation, scaling, offsetting etc.). Global Mapper has a tight non-lossy grid format (.gmg).

Global Mapper also offers display and logging of GPS data, draping of imagery or maps on grids with 3-D display, digitization tools, contouring, view-shed and water-shed analysis, screen capture, topographic profiles with slopes, palette construction, shading options etc. It has also become an important connectivity and display tool for sophisticated multi-platform users of programs such as GMT, ArcGIS, CARIS, and IVS Fledermaus. Two features lacking on Global Mapper are the ability to convert .dxf files to xyz (which can be done with the free program dxf2xyz from www.guthcad.com), and the use of degrees minutes and seconds for geographic coordinates output to .csv format. This is important for merging new digitized contours and shorelines with the spot soundings digitized years ago, to form a single file for interpolation by Surfer. The work-around of course is to use Cartesian meridional parts for Mercator maps, or else UTM coordinates.

4.4.3 Golden Software Inc. Surfer® Version 9

A license for Surfer's latest version 9 is $699. Global Mapper has the 3-D display, contouring and hypsometric shading and palette selection capabilities of Surfer. However Surfer is the program of choice for interpolation of grids from the tens of thousands of points generated by r2v and the historical digitization of soundings. The Surfer gridding interface allows rapid investigation of the input files, and the Filter Data function deals with duplicates as well as averaging points that are so close that steep gradients would be formed.

Grid Construction

As an example, over 55% of the Mediterranean has now been mapped with multibeam sonar. The MediMap Group (http://www.ciesm.org/marine/morphomap.htm) has pooled their data and it is currently available within the group at a node spacing of 500 m. Efforts to decrease this spacing has apparently been opposed by commercial interests. Basically this coverage is sufficiently good to make a first pass at a 0.1' grid for IBCM-II. Israel is now concluding a decade of pro-active mapping with a Kongsberg-Simrad EM1002 owned by John Hall, installed on the Israeli government vessel R/V Etziona. Over 10,000 km² of the Mediterranean offshore have been surveyed, from minimum depths of 15 m to about 700m with the EM1002, and then recently out to 1,600 m depth with a rented L-3 ELAC Gmbh SB3050 system. While the Israeli government regards the digital data as classified, grids of 100 m on the shelf and 50 m in deeper waters are available. Hopefully this will encourage other countries to allow use of higher definition deep water data.

The seas that are of interest follow, in order of size etc.:

Water Body	Total Area	Max Depth	Shoreline	Countries
Mediterranean	2,966,000 km²	5267 m	46,000 km	22
Red Sea	438,000 km²	2211 m	11,254 km	8
Black Sea	474,000 km²	2206 m	4,340 km	7
Caspian Sea	371,000 km²	1025m	6,397 km	4
Persian Gulf	251,000 km²	90 m	3,334 km	8

Russian charting at about 1:200,000 covers the periphery of all these bodies. Thus grids can be obtained by first taking the land coverage at 30 m from the latest ASTER-GDEM. These come in one degree squares (called granules) and can be downloaded, up to 2000 at a time, from http://wist.echo.nasa.gov/api/. It is necessary to register, and after choosing and verifying the coverage wanted, one receives a verification and link to an ftp site. The granules can be quickly downloaded with an ftp client program like Filezilla (http://filezilla-project.org). The files consist of the height values (.dem format) and the number of scenes used (.num). Global Mapper 12 will rapidly ingest the height data while ignoring the .num component, and can then output single .gmg grid files with the data for the overall area. The next step is to remove the data over the sea. This can be done by using Tools>Control Center> Options>Alter Elevation Values to reject anything below 0.1 m. While this will generally work, sometimes the alternative is to single out specific granules and crop them using an area boundary in Global Mapper or a blanking file in Surfer. Such a file is usually the 0 m contour from the digitizing done in r2v.

The clipped ASTER-GDEM file gives a pretty good indication of where the land is, and for checking the marine data which often is pre-WGS84 or WGS72 datums. Older Russian charts sometimes referred to the Russian Pulkovo S42 datum of 1942, or in the case of early British surveys, relative to one of the 700 datums used in UKHO surveys. Global Mapper's ability to overlay and translate charts can correct some of these geodetic displacements.

The next step is to geo-reference the 300 dpi color scans of the charts, one at a time, using Global Mapper. The georeferencing can be checked by bringing up Tools>Configure>General>Grid Display>Lat/Long Grid and specifying the map graticule interval. Visual inspection will then indicate the quality of the georeferencing. General four corner control points will suffice, but for a long body like the Red Sea, perhaps as many as 20 points may be necessary. Using File>Export Raster Image Format>GeoTiff allows export of a GeoTiff with .tfw true world file that can be used in r2v. At present 92 different projections can be used. Generally it is best to use Cartesian coordinate like meridinial parts or UTM.

Within r2v the coastline and available contours are digitized and their depths noted. Where inset charts at larger scale are available, the spot soundings were separately digitized, and data from them can be digitized separately and then later merged by loading the larger scale .dxf contour file into the smaller scale chart. Where necessary, spot soundings can also be picked up. In order to seamlessly place the bathymetry under the land coverage, the high elevations shown on the navigational chart may be digitized so that the kriging process will apply a landward slope commensurate with the actual topography. Inshore contours at 3, 6, 10, 20, and 50 m are often almost continuous, so that the shallows are reasonably well represented. For the southern Red Sea, where there are numerous reef areas indicated from space imagery, fictitious contours at 20 and 3 m depths help to define the reefs. In general, where ships travel, the hydrographers have soundings, but of course, regardless of what the final grid spacing is, the results must be clearly marked as "Not To Be Used For Navigation."

Coastal navigational charts generally overlap one another, so if each is digitized and then interpolated to a grid, their grids can later be cropped and stacked appropriately within Global Mapper for output as the final grid. In the case of the Red Sea, UTM grids were interpolated at 100 m for Zones 36 and 37. Global Mapper keeps the grids at the finest scale used, so with ASTER-GDEM 30 m data the default output would be 100 m.

REFERENCES

Carron, M. J., P. R. Vogt, and W.-Y. Jung. 2001. A proposed international long-term project to systematically map the world's ocean floors from beach to trench: GOMaP (Global Ocean Mapping Program). Int. Hydrographic Rev., 2, p. 49–55.

Doytsher, Y., and Hall, J. K., 1997. Gridded affine transformation and rubber-sheeting algorithm with FORTRAN program for calibrating scanned hydrographic survey maps. Computers & Geosciences, 23(7), p. 785-791.

Hardy, R. L., 1971. Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res., 76(8), p. 1905-1915.

Hall, J. K., Schwartz, E., and Cleave, R. L. W., 1990. The Israeli DTM (Digital Terrain Map) Project. In: Microcomputer Applications in Geology, II (J. T. Hanley and D. F. Merriam, eds.). pp. 111-118.

Computers and Geology, Vol. 6, Pergamon Press, New York. Computer program on diskette as part of the COGS (Computer Oriented Geological Society) public-domain library.

Hall, J. K., Udintsev, G. B. and Odinokov, Yu. Yu., 1994. The bottom relief of the Levantine Sea. In: Geological structure of the North-eastern Mediterranean (Cruise 5 of the Research Vessel 'Akademik Nikolaj Strakhov'). (V. A. Krasheninnikov and J. K. Hall, eds.), p. 5-31. Historical Productions-Hall, Jerusalem, April 1994.

Vogt, P. R., W. -Y. Jung, and D. J. Nagel, 2000. GOMaP: A matchless resolution to start the new millennium. Eos Transactions, American Geophysical Union, 81(23), p. 254, 258. June 6, 2000.

Chapter 5.0 Gathering Data

There are many types of geophysical data available for free download from the internet. In this chapter we show where to obtain data useful for researchers desiring to create grids from publicly available bathymetric data.

5.1 NGDC and IHO DCDB Bathymetry Data

Contributed by Susan McLean, National Geophysical Data Center, USA

The National Geophysical Data Center's (NGDC) mission is to provide long-term scientific data stewardship for marine geology and geophysical data, ensuring quality, integrity, and accessibility. This includes archiving and making available bathymetric data collected by NOAA, UNOLS (University-National Oceanographic Laboratory System), other US government agencies, and foreign entities. These data include ocean multibeam and single beam surveys, coastal multibeam and single beam surveys (hydrographic surveys), and digital elevation models compiled from various data sources. NGDC is co-located with the International Hydrographic Organization Data Center for Digital Bathymetry (IHO DCDB) which serves IHO member states and the international community by archiving and providing access to global bathymetric data. Both Centers provide ancillary data files essential to bathymetric data processing. These files include metadata records, sound velocity profiles, navigation files, tides files, and ship logs. When provided by the chief scientist, survey derived products are also made available. However, the bulk of data archived and available to the public are raw multibeam files.
Availability of the raw data allows scientists to edit the data to any specification. Raw multibeam data require a multibeam reader/editor to produce products which are more widely interoperable. Since manipulating the raw data files can be time consuming, the centers offer products which are already edited and in a format readable by many desktop applications. One such product is the Bathymetric Attributed Grid (BAG) file. BAG (http://www.opennavsurf.org) is a gridded, multi-dimensional bathymetric data file.

5.1.1 Discovering and Identifying Data

NGDC's Bathymetry Data Viewer (http://maps.ngdc.noaa.gov/viewers/bathymetry) (Figure 5.1) is an interactive mapping application that allows discovery, identification, and access to bathymetry data archived at NGDC. Here we describe how to interact with the map to identify and obtain data from the NGDC website.

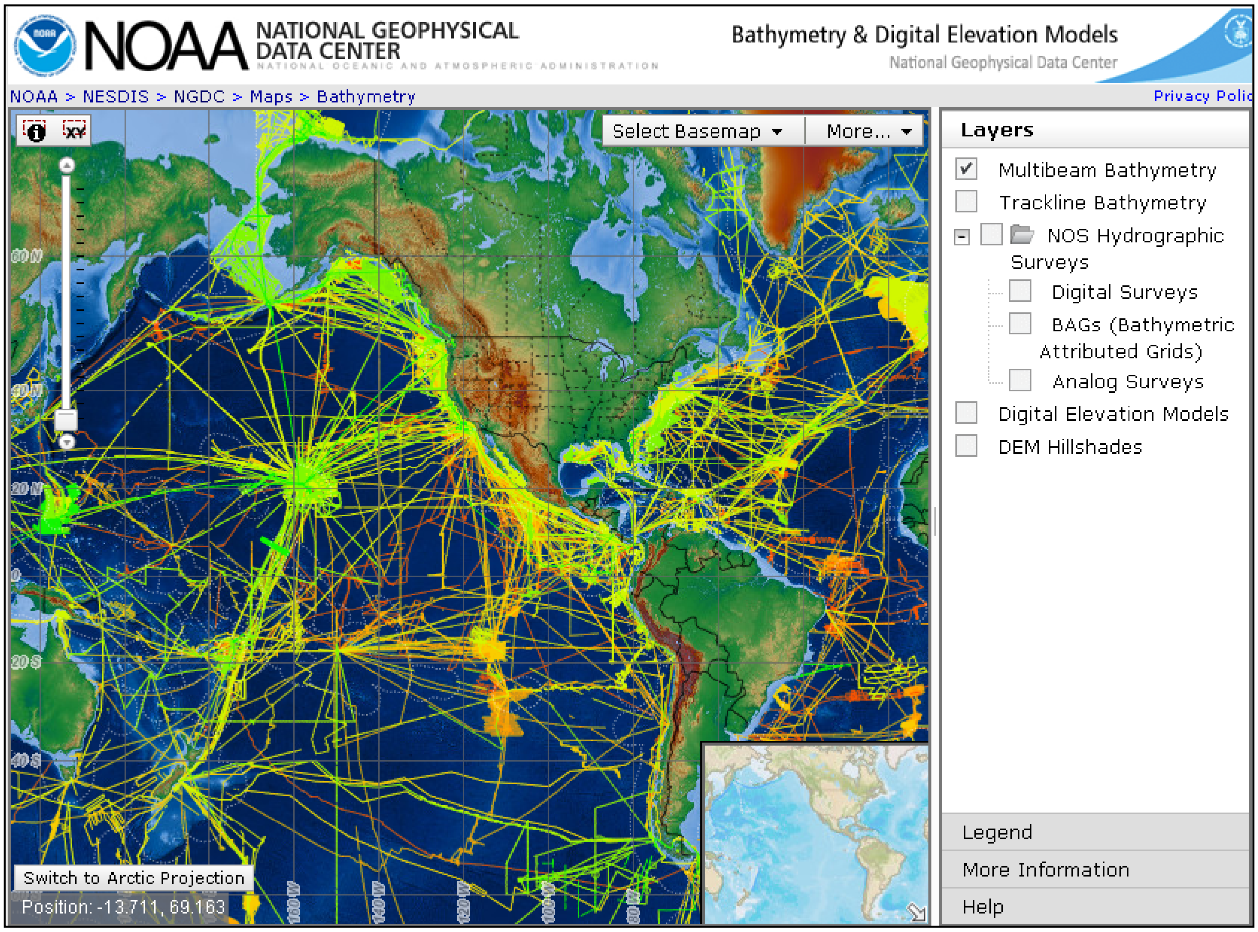


Figure 5.1 NGDC Bathymetry Data Viewer, displaying Multibeam Bathymetry Surveys. Surveys are color-coded by year (newer surveys are green and older surveys are orange).

Navigating the map:

Click and drag to pan the map.
Use the zoom slider to zoom in/out, or shift-click and drag to zoom into a box.
Toggle the different basemaps with the "Select Basemap" drop-down menu.
Toggle the boundaries/labels and graticule with the "More…" drop-down menu.
Switch to an Arctic map projection with the button in the lower-left.
Toggle the various data layers with the checkboxes in the right panel. The available layers are:
Multibeam Bathymetry surveys (shown in Figure 5.1)
Trackline (single-beam) Bathymetry surveys (shown in Figure 5.2)
NOS Hydrographic Surveys for U.S. near-shore areas. The user can toggle Digital Surveys, Analog Surveys, and Bathymetric Attributed Grids (shown in Figure 5.3)
Digital Elevation Models (shown in Figure 5.4)
DEM Hillshades: shaded-relief visualization of DEMs (also shown in Figure 5.4)

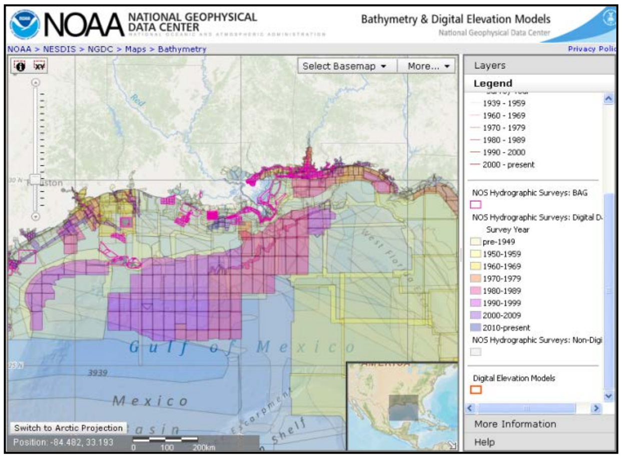


Figure 5.2 Trackline Bathymetry Surveys

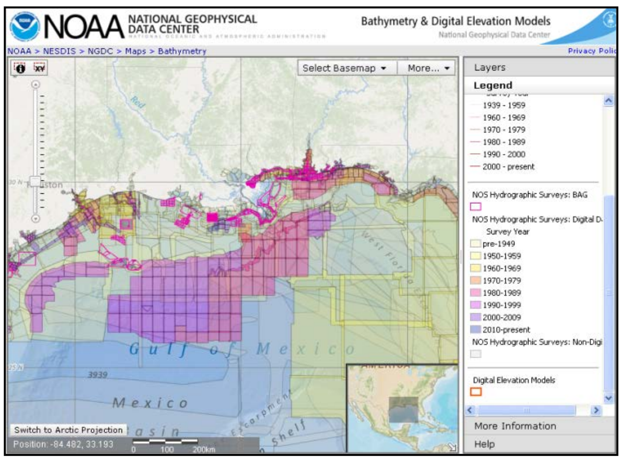


Figure 5.3 NOS Hydrographic Surveys

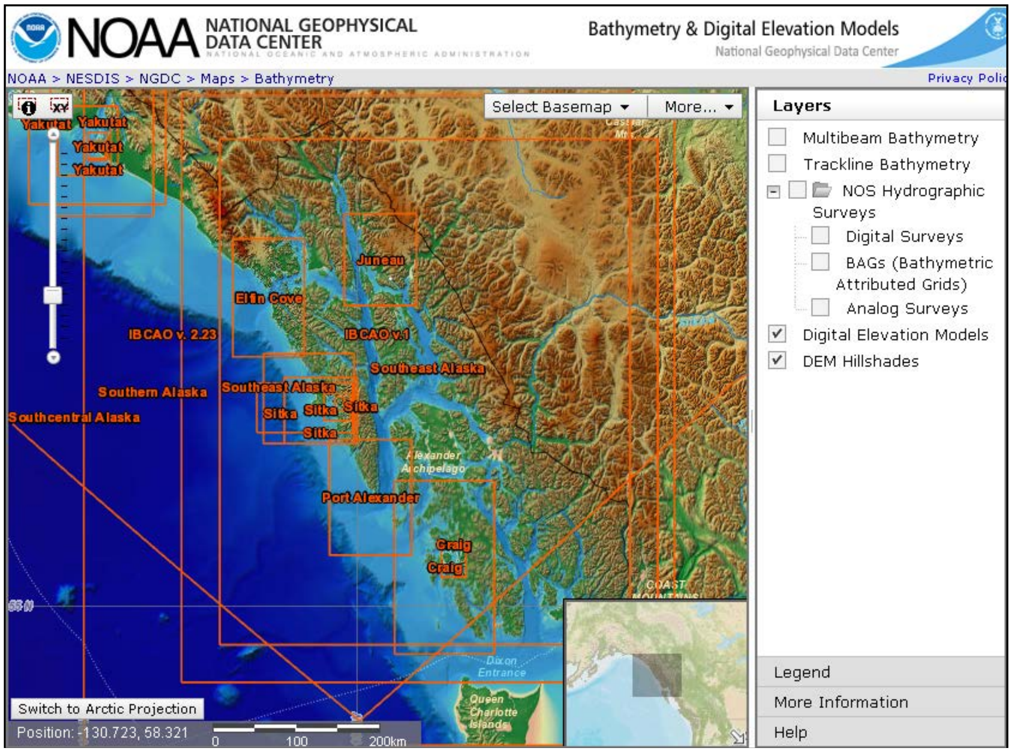


Figure 5.4 Extents of Digital Elevation Models, with shaded-relief visualizations.

Identifying data:

Click on the map to identify data at the point clicked.
Alternatively, a user may define a rectangular area using the tool to draw a box on the map, or the tool to define a latitude/longitude extent.
A popup will appear, displaying data near that location (Figure 5.5).
Multiple surveys may be returned near where user clicks; mouse over the list to highlight the surveys on the map.

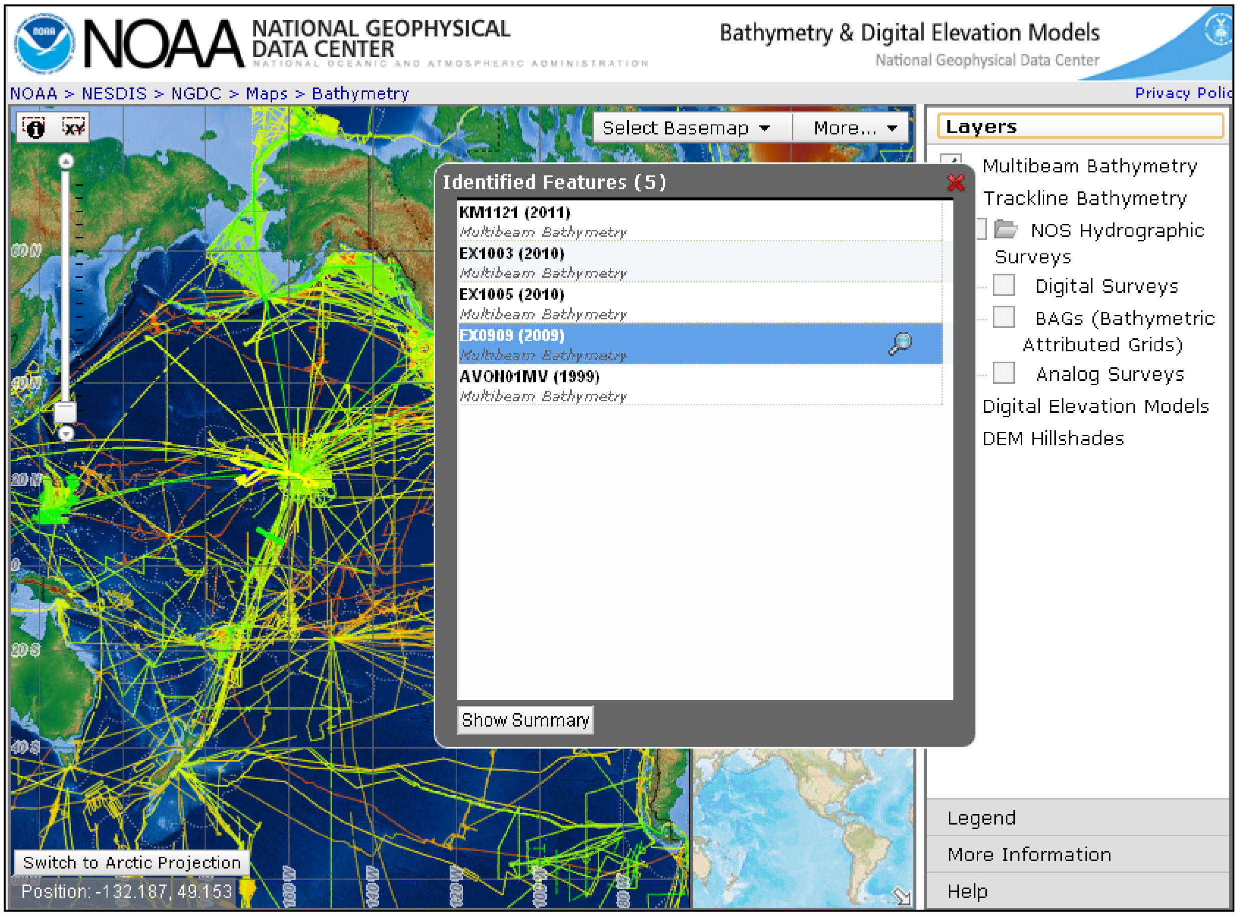


Figure 5.5 After clicking near the Hawaiian Islands, 5 surveys have been identified. EX0909 is highlighted on the map.

Zoom to the extent of a survey using the magnifying-glass icon.
Click "Show Summary" to view a summary of all surveys in the list.

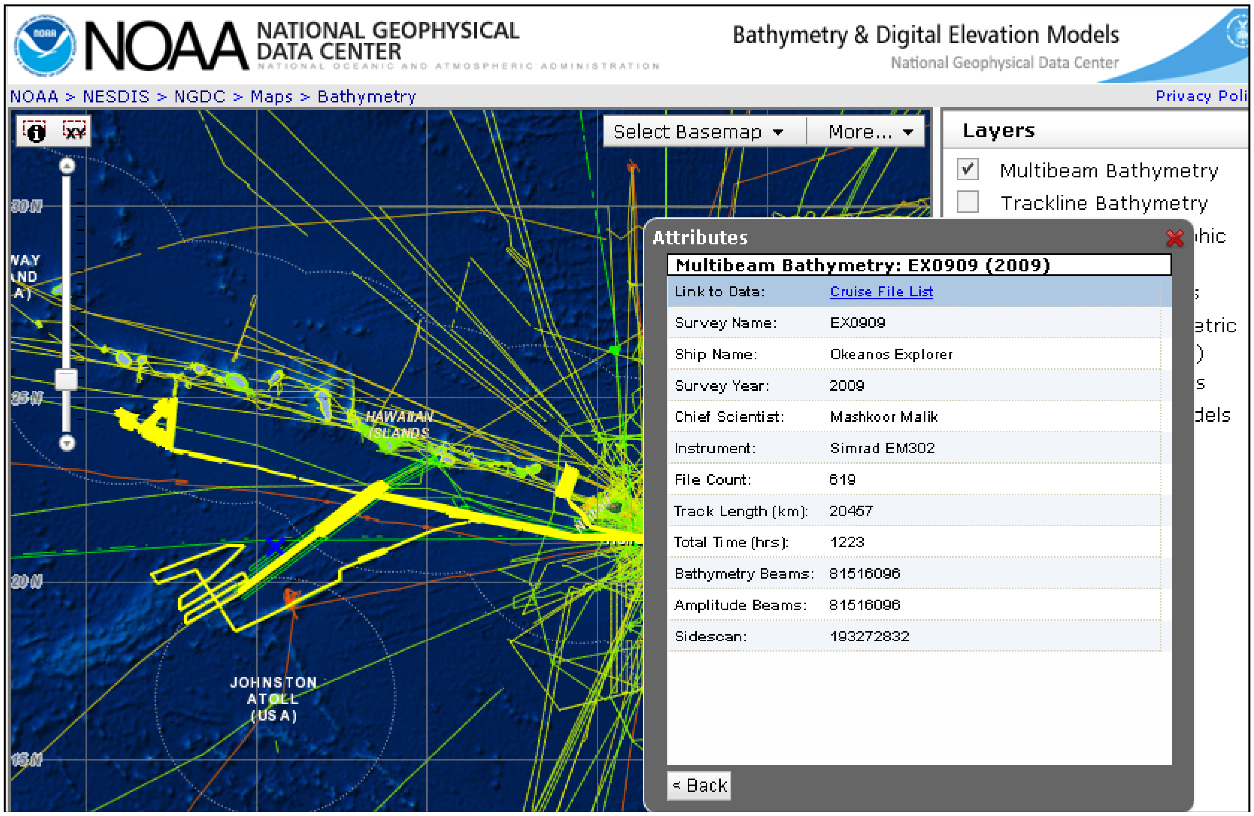


Figure 5.6 Attributes shown for survey EX0909. For multibeam surveys, click on "Cruise File List" to go to a page where data can be downloaded (described below).

Click on a row to display additional attributes for that survey (Figure 5.6).
A user can follow the link to go to a page where the data can be downloaded.

Downloading Data:

After clicking the download link from the map, a list of Multibeam data files is shown (Figure 5.7) and can be downloaded either individually, or as one large package. Shown below are download pages for NOS Hydrographic Surveys (Figure 5.8), Trackline Bathymetry (Figure 5.9), and Digital Elevation Models (Figure 5.10).

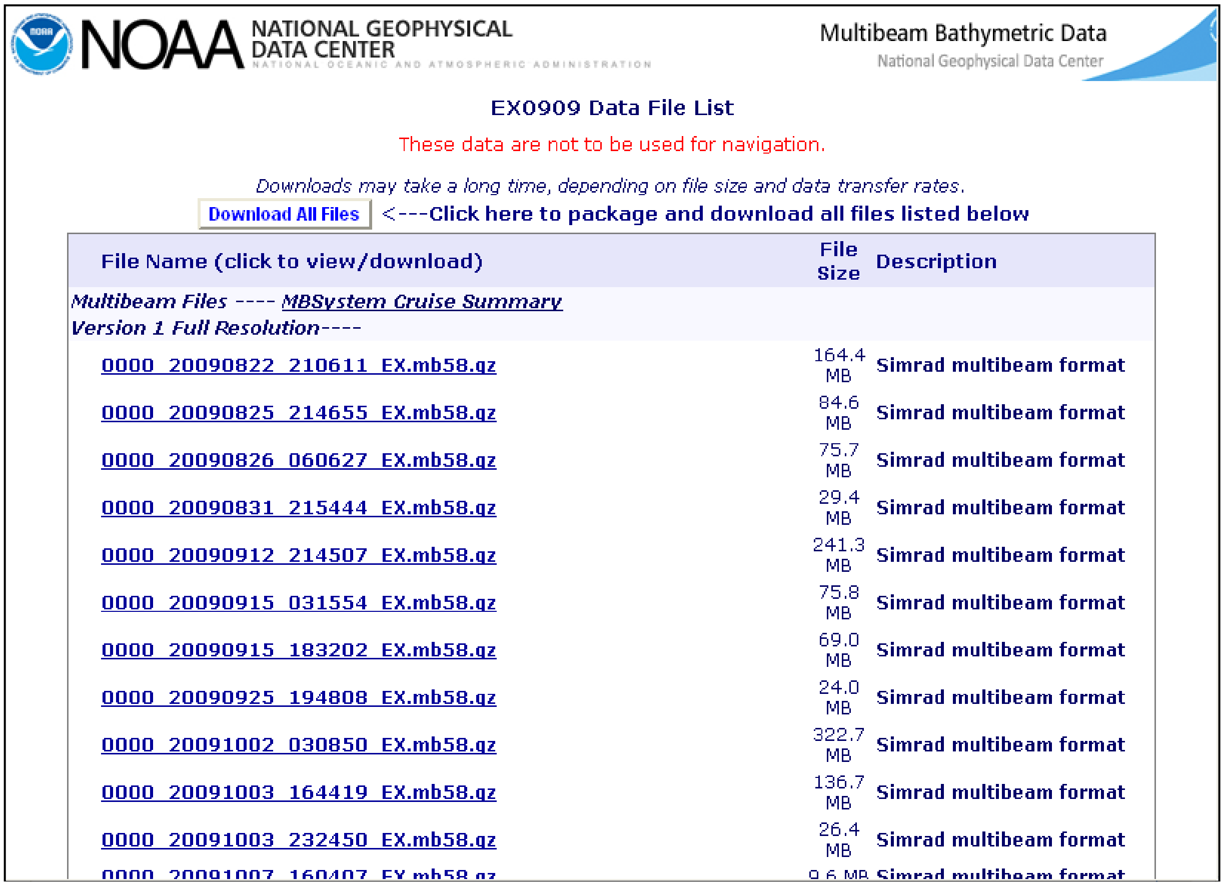


Figure 5.7 Multibeam Data download page.


NOAA NATIONAL GEOPHYSICAL DATA CENTER
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Office of Coast Survey
NOAA National Ocean Service

Survey Product List

These data are not to be used for navigation.
For navigation please refer to NOS Nautical Charts.

Downloads may take a long time, depending on file size and data transfer rates.

File Name (click to view/download)	File Size	Description
FGDC Metadata ----		
gov.noaa.nos:H11272		**NOAA/NOS FGDC/RSE metadata record** Usually presented as an XML document, which captures the basic characteristics of a data or information resource.
Descriptive Report ----		
H11272.pdf	3.1 MB	**NOAA/NOS Descriptive Report in PDF format** The report may be viewed using a free Adobe Reader.
Bathymetry Attributed Grid (BAG) ----		
Select / Unselect All BAG files for conversion to XYZ text format. Check here to include Uncertainty Data [Select Data]		
H11272_10m_Combined_MLLW_3of3.bag.gz	12.4 MB	**NOAA/NOS Bathymetry Attributed Grid (BAG) file** The BAG is a gridded, multi-dimensional bathymetric data file. (http://www.opennavsurf.org/). Free Viewer
H11272_10m_MLLW_2of3.bag.gz	10.2 MB	**NOAA/NOS Bathymetry Attributed Grid (BAG) file** The BAG is a gridded, multi-dimensional bathymetric data file. (http://www.opennavsurf.org/). Free Viewer

Figure 5.8 NOS Hydrographic Survey download page. BAG files are available for this survey.

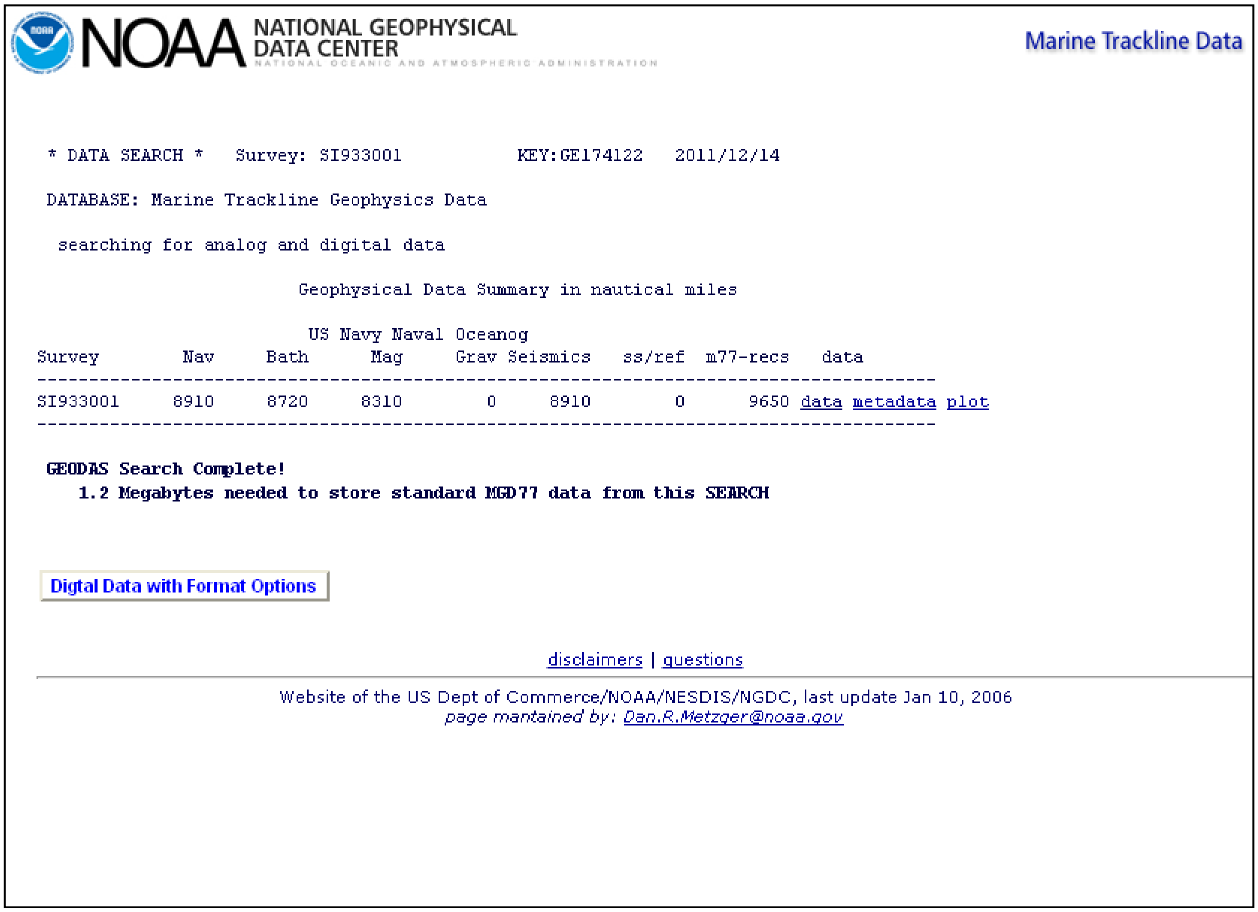


Figure 5.9 Trackline Bathymetry data download page (GEODAS).

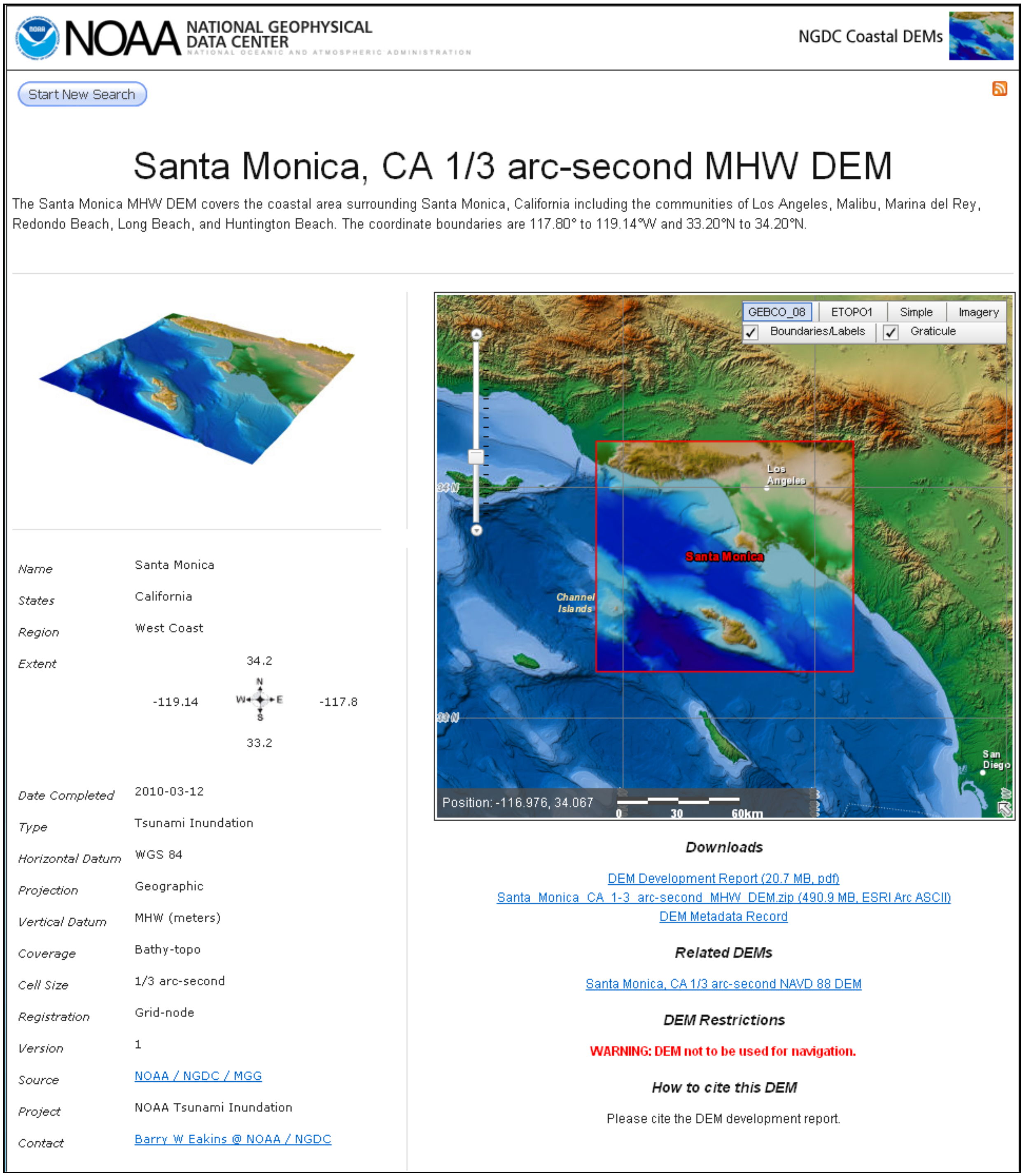


Figure 5.10 DEM Download Page. Data is provided in ESRI Arc ASCII format.

5.2 JAMSTEC Data

Contributed by K. M. Marks, NOAA Laboratory for Satellite Altimetry, USA

Many types of geophysical data are available for download from the JAMSTEC website (http://www.godac.jamstec.go.jp/cruisedata/e/), which is shown in Figure 5.11. Users who make use of the JAMSTEC website are requested to register on https://www.godac.jamstec.go.jp/cruisedata/e/userRegistS.html. Here we demonstrate how to obtain multibeam data from the JAMSTEC website.

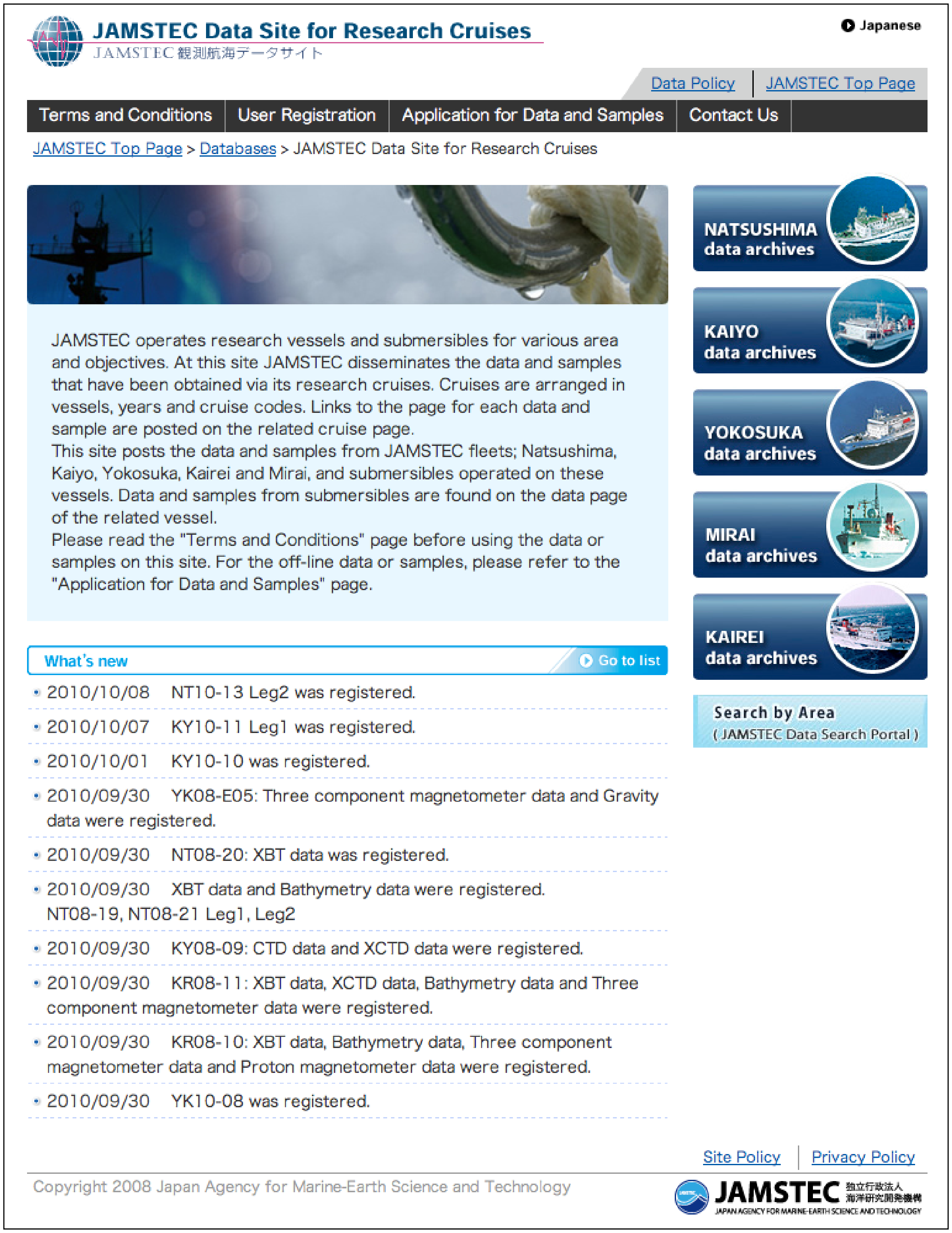


Figure 5.11 JAMSTEC Data Site for Research Cruises.

Multibeam data may be downloaded from the JAMSTEC website as follows. The user first selects the "data archives" link for one of the five research vessels on the main JAMSTEC website (Fig. 5.11). The link opens a global map of cruises color-coded by collection year (Fig. 5.12). The user selects the desired year, and then the cruise ID. Selecting "Apply" opens the individual cruise web page (Fig. 5.13). To reach the bathymetry web page, the user selects "Bathymetry" from the right side of the cruise web page (see Fig. 5.13).

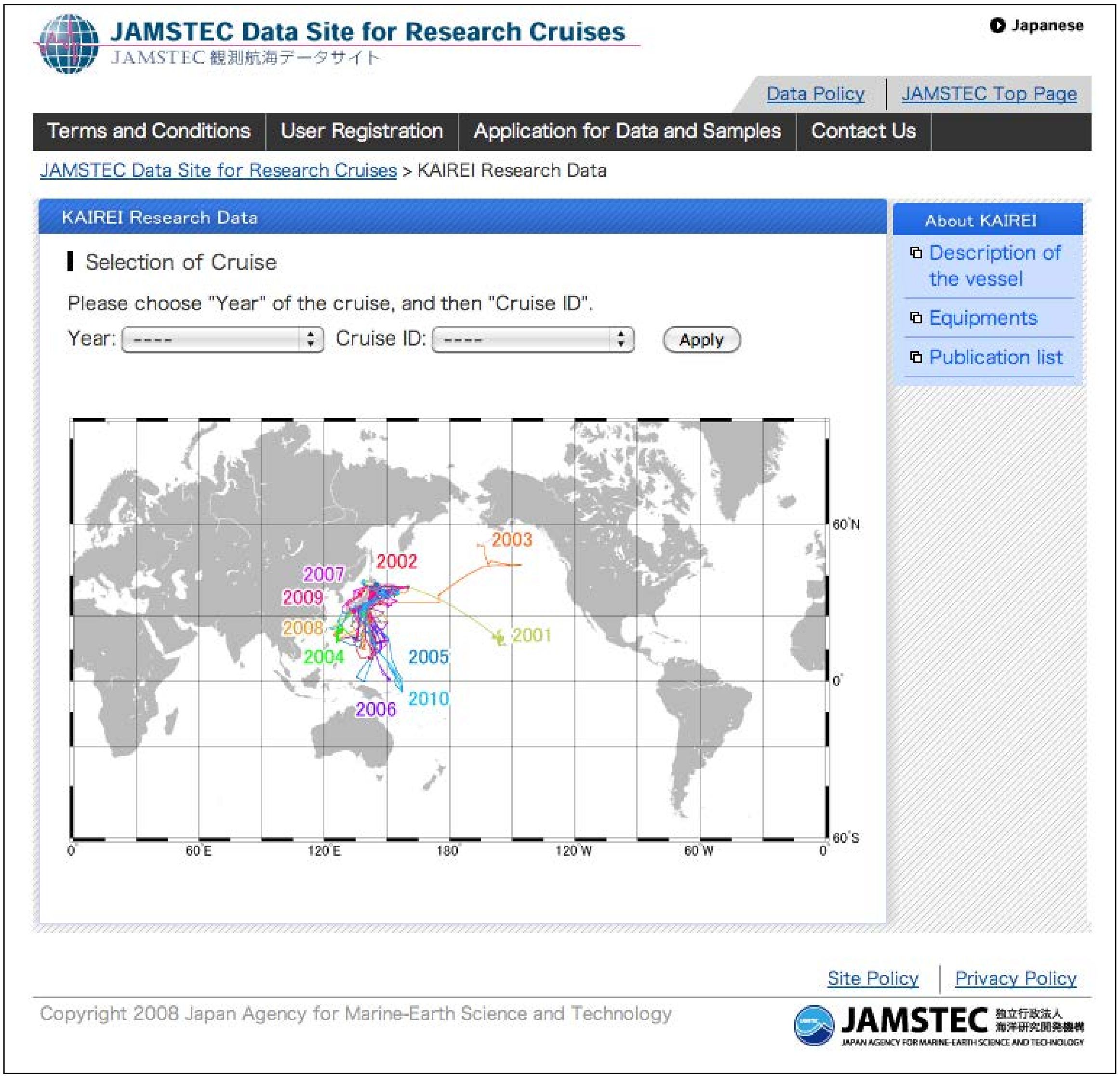


Figure 5.12 KAIREI Research Data web page.

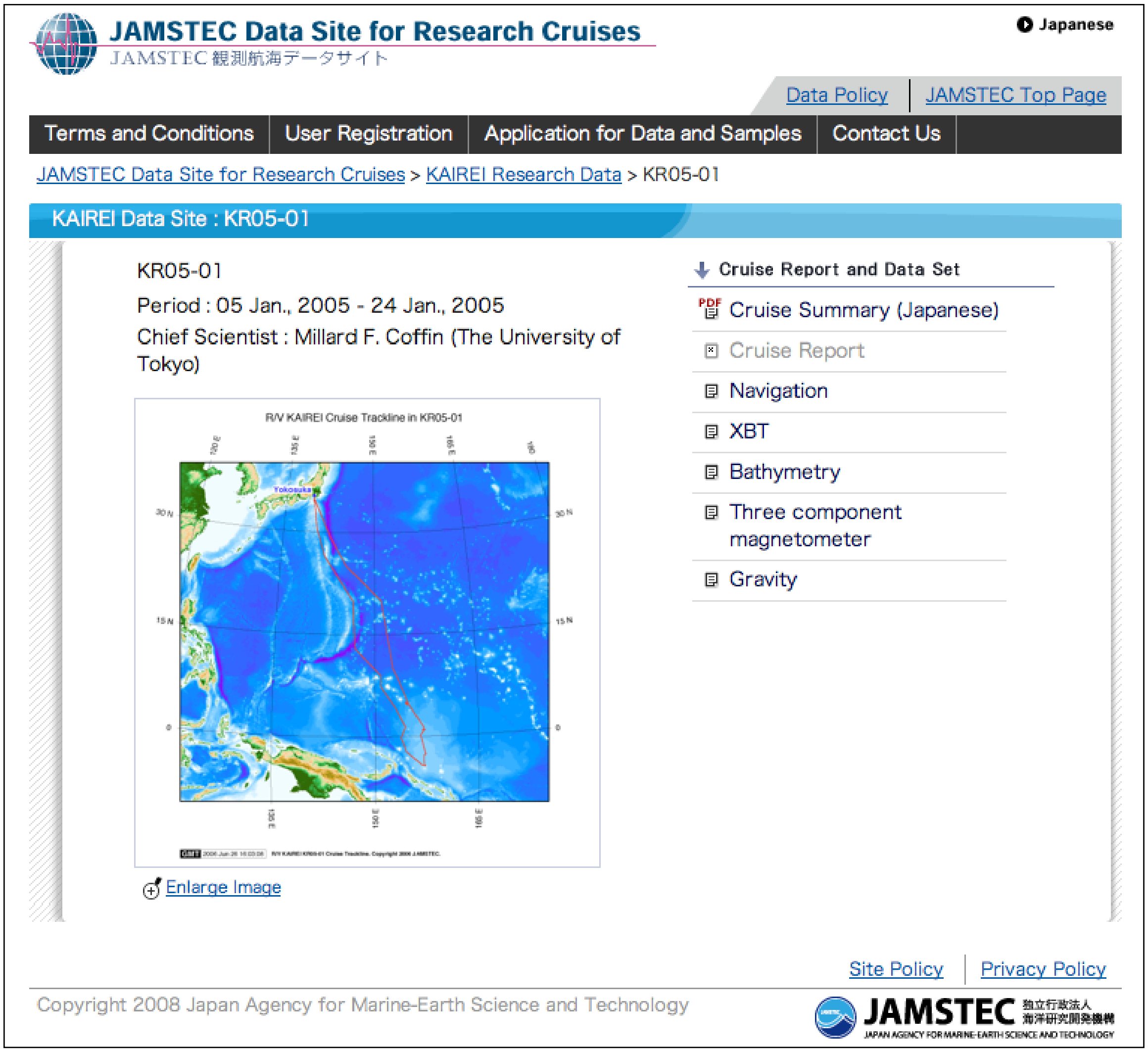


Figure 5.13 KR05-01 Cruise web page.

Bathymetry data may be downloaded from the bathymetry web page (Fig. 5.14) via the "Data" link, as well as information on the instruments, data format, and collection system details (i.e., the "Instruments," "Format description," and "Readme" links, respectively). The bathymetry data may be downloaded as a number of compressed files in "zip" format (see Fig. 5.15) that need to be uncompressed ("unzipped") and then assembled sequentially into a single file. The bathymetry data are ASCII *xyz* (longitude, latitude, depth) multibeam echo sounder ping records. There may not be data along the entire track plot shown on the bathymetry web page and making a plot of the coverage will reveal any gaps.

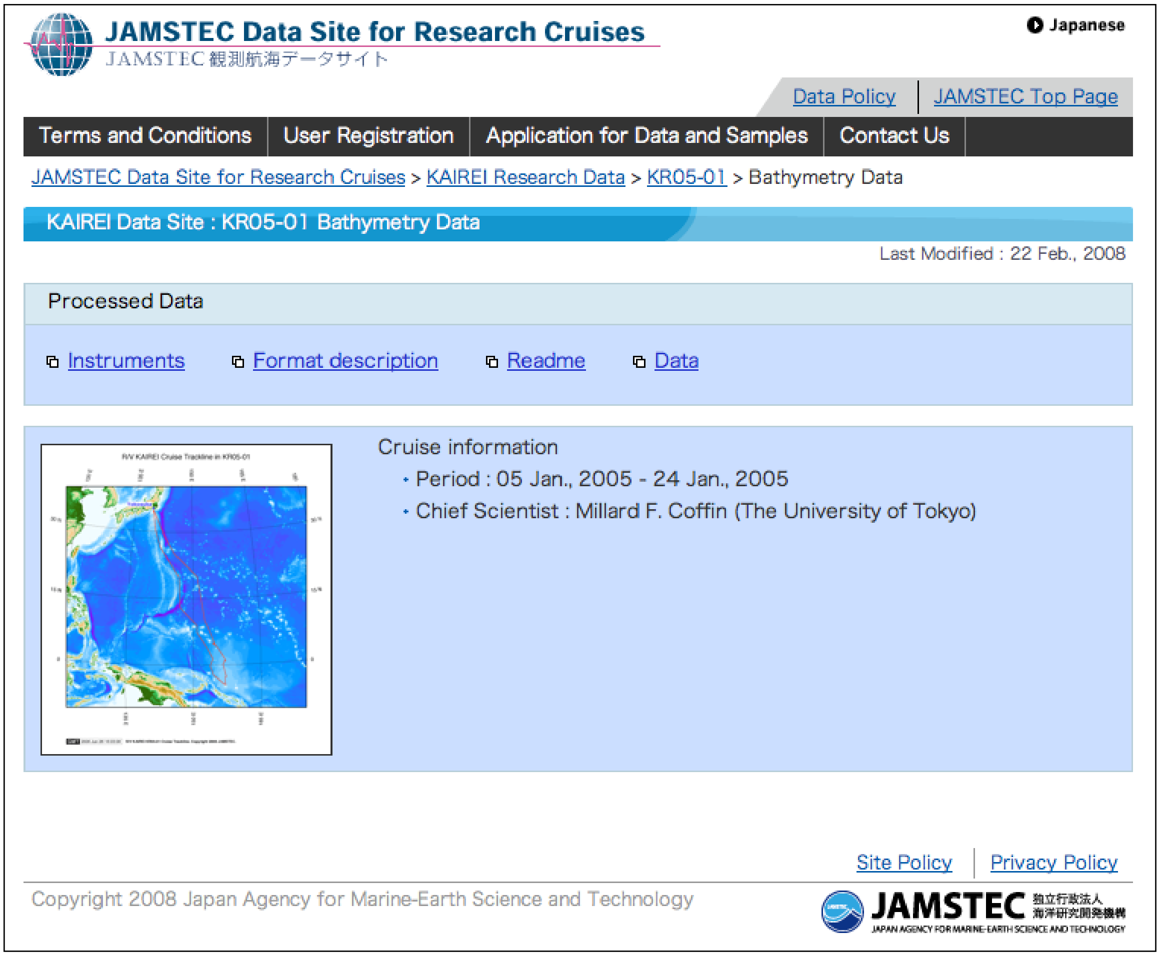


Figure 5.14 KR05-01 Bathymetry Data web page.

JAMSTEC Data Site for Research Cruises
JAMSTEC 観測航海データサイト

Japanese

Data Policy | JAMSTEC Top Page

Terms and Conditions | User Registration | Application for Data and Samples | Contact Us

JAMSTEC Data Site for Research Cruises > KAIREI Research Data > KR05-01 > Bathymetry Data > Processed Data

KAIREI Data Site : KR05-01 Bathymetry Data : Processed Data

- Data Files

File name : YYYYMMDD.dat
YYYY: year, MM : month, DD: day
* These data are compressed in zip format, please use that after unpacking.

File name	Size	Date
20050106.zip	0.14 MB	06 Jan., 2005
20050107.zip	3.76 MB	07 Jan., 2005
20050108.zip	4.91 MB	08 Jan., 2005
20050109.zip	3.86 MB	09 Jan., 2005
20050110.zip	4.23 MB	10 Jan., 2005
20050111.zip	6.71 MB	11 Jan., 2005
20050112.zip	7.37 MB	12 Jan., 2005
20050113.zip	8.67 MB	13 Jan., 2005
20050114.zip	9.06 MB	14 Jan., 2005
20050115.zip	11.38 MB	15 Jan., 2005
20050116.zip	10.46 MB	16 Jan., 2005
20050117.zip	6.63 MB	17 Jan., 2005
20050118.zip	5.77 MB	18 Jan., 2005
20050119.zip	2.51 MB	19 Jan., 2005

Site Policy | Privacy Policy

Copyright 2008 Japan Agency for Marine-Earth Science and Technology

JAMSTEC 独立行政法人 海洋研究開発機構
JAPAN AGENCY FOR MARINE-EARTH SCIENCE AND TECHNOLOGY

Figure 5.15 KR05-01 Bathymetry Data files available for download.

An alternate way to reach bathymetry web pages is for the user to click on the “JAMSTEC Data Search Portal” tool (see Fig. 5.11). This accesses instructions on how to use the tool as well as provides an entryway to the tool’s ArcIMS interface (see Fig. 5.16). On the ArcIMS interface, the user may drag the red box on the global map (upper left corner) to the desired study location, then select “Quick Search” to drag a box on the main map. This brings up a search result panel (see Fig. 5.17) that lists the cruises that traverse the study box. The bathymetry web page for the cruise of interest can then be reached by clicking on “Jump.”

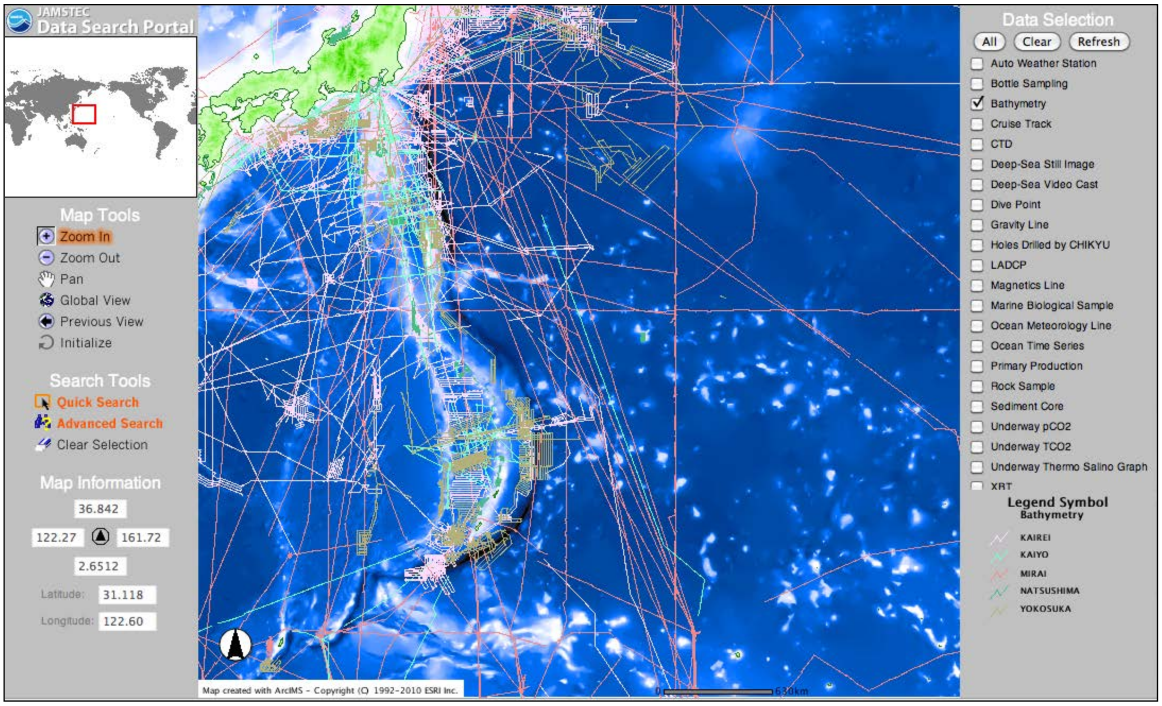


Figure 5.16 JAMSTEC Data Search Portal.

Search Result (Select Data Set : Bathymetry) 16 Selected

No.	Data ID	Latitude	Longitude	Period (start)	Period (end)	Ship Name	Cruise ID	DSRV Name	Dive No.	Web Site
1	0013-00000027			2002/02/20	2002/03/29	MIRAI	MR02-K02			Jump
2	0013-00000035			2003/02/19	2003/03/29	MIRAI	MR03-K01			Jump
3	0013-00000052			2004/11/16	2004/12/08	MIRAI	MR04-07			Jump
4	0013-00000054			2005/01/13	2005/02/18	MIRAI	MR04-08_leg2			Jump
5	0013-00000056			2005/05/24	2005/06/30	MIRAI	MR05-02			Jump
6	0013-00000064			2006/02/04	2006/03/17	MIRAI	MR06-01			Jump
7	0013-00000073			2007/02/15	2007/03/25	MIRAI	MR07-01			Jump
8	0013-00000075			2007/05/30	2007/07/13	MIRAI	MR07-03			Jump
9	0013-00000085			2008/07/01	2008/08/05	MIRAI	MR08-03			Jump
10	0013-00000211			2005/01/04	2005/01/23	KAIREI	KR05-01			Jump
11	0013-00000228			2005/12/09	2005/12/24	KAIREI	KR05-17			Jump
12	0013-00000242			2006/09/11	2006/09/22	KAIREI	KR06-12			
13	0013-00000264			2007/11/22	2007/12/01	KAIREI	KR07-16			Jump
14	0013-00000326			2002/09/29	2002/10/14	KAIYO	KY02-10_leg1			
15	0013-00000639			2001/01/03	2001/01/14	YOKOSUKA	YK01-01			
16	0013-00000915			2009/11/02	2009/12/11	MIRAI	MR09-04			

Figure 5.17 Data Search Portal Results.

Chapter 6.0 Data Cleaning

6.1 Quality Assessment

Contributed by Barry Eakins, NOAA-CIRES, Rob Hare, Canadian Hydrographic Service, and Martin Jakobsson, Stockholm University, Sweden

6.1.1 Sources of uncertainty

Elevation value

Instruments such as multibeam swath sonars and airborne lidar do not actually measure elevation, but rather travel time (of sound or light, respectively). Many different sources therefore contribute to the uncertainty of a calculated depth/elevation value. Although the following text focuses on uncertainties in data acquired with multibeam swath sonar systems, it should be noted that it is in principle applicable to data acquired with any elevation measurement system; a single-beam echosounder is just a special case of a multibeam echosounder (i.e. nadir beam only). Agencies conducting surveys are encouraged to provide uncertainty budgets, or total propagated uncertainty (TPU) estimates for their own systems.

i. *Sound or light speed uncertainty – how accurately was sensor measurement or time converted into a depth/elevation value?*

   Multibeam swath sonar systems rely on frequent measurements of sound speed, typically taken an XBT (expendable bathythermograph) or CTD (conductivity-temperature-depth package), and sometimes measured directly using an SVP (sound speed profiler). These measurements should be taken at least daily, or with changing weather conditions, as longer time intervals decrease the accuracy of the soundings due to improper ray-tracing. Measurements should be taken more frequently in areas of known oceanographic fronts and off the mouths of major rivers.

   The sound speed profile is used to calculate depth for each beam, via ray tracing, and is embedded in the binary sonar files. The profiles can be extracted and replaced with different profiles using multibeam processing software such as MB-System, which can recalculate depths and positions for each beam footprint.

   Because sound travels farthest for the outermost, lowest grazing angle beams, especially in deep water, and because ray bending increases as the tangent of the beam angle, these beams have the largest depth uncertainty due to sound speed uncertainty caused by ray bending, and often contain anomalous values inconsistent with other, overlapping data. As such, they are frequently flagged for exclusion. If sound speed profiles cannot be taken at a rate which minimizes the uncertainty due to refraction, then line spacing should be decreased in order to increase the overlapping region, and provide additional confidence in the bathymetry with many of the outer beams eliminated from the gridding process.

   Uncertainty in the measurement of vessel roll with also contribute significantly to the TPU of depth in the outer beams.

ii. *Position uncertainty – how well known was the position of the sensor?*

With swath sonar systems the distance between the sounding on the seafloor and the positioning system antenna can be very large, especially in deep water. Because of this, sounding position uncertainty is a function of the errors in vessel heading, beam angle solution, refraction correction model and the water depth, in addition to the uncertainty of the positioning system itself.

Roll and pitch errors will also contribute to the uncertainty in the positions of soundings. Overall, it may be very difficult to determine the position uncertainty for each sounding as a function of depth. The uncertainties are a function not only of the swath system but also of the location of, offsets to and accuracies of the auxiliary sensors.

The use of non-vertical beams introduces additional uncertainties caused by incorrect knowledge of the ship's orientation at the time of transmission and at time of reception of sonar echoes.

Uncertainty associated with the calculated depth of an individual swath sonar beam will include the following components:

a) Positioning system uncertainty (e.g. of the GPS antenna)
b) Sensor location uncertainty (of the sonar's transducer and hydrophones relative to the GPS antenna)
c) Beam angle uncertainty (relative to the sensor), including beam steering uncertainty when sound speed at the transducer face in not directly measured
d) Uncertainty associated with beam detection of seabed reflection echo (angle or time)
e) Uncertainty in the orientation of the vertical motion sensor (relative to the sonar); pitch bias and roll bias
f) Vessel motion sensor uncertainty, i.e. roll, pitch and heave
g) Uncertainty in vessel heading (distinct from path over ground)
h) Uncertainty associated with the ray path model (including the sound speed profile)
i) Sensor offset uncertainty (relative to local water level)
j) Ellipsoidal / vertical datum transformation model uncertainty (if applicable)
k) Time synchronization / latency between all of the systems

All uncertainties should be combined statistically to obtain a *total vertical uncertainty* (TVU).

Morphologic change since data collection

Morphologic change is an ongoing process that will, over time, decrease the accuracy of survey data. In the coastal zone especially, the movement of sand alters seabed morphology. This may occur as the result of a major storm, or long-term sediment dispersal or accumulation. Regional tectonic uplift or subsidence may also play a role. The older the data is the greater its uncertainty, with the greatest rates of change occurring in shallow water.

Data not measuring ground surface

Not all instrument measurements detect Earth's solid surface. Many individual sonar depths may actually reflect water column noise. Spikes often occur directly along the ship track, where the sound energy returning to the transducer is strongest. Returns from the water surface or from tree canopy are also common in topographic lidar. These values need to be identified and deleted.

6.1.2 How to determine data uncertainty

a. Typical for modern data types
   i. Lidar [*Fugro Earthdata*, 2007]

Point spacing	Vertical accuracy (RMSE)
1 m	0.09 m
2 m	0.20 m
3 m	0.30 m
4 m	0.40 m
5 m	0.51 m

   ii. Multibeam swath sonar
   iii. Hydrographic surveys undertaken prior to 1998

Depth	Vertical accuracy
< 30 m	0.3 m
> 30 m	1% of depth

b. Estimating navigation uncertainty for older data
c. Comparison with other data

6.2 Data Cleaning Techniques

Contributed by Rob Hare and Paola Travaglini, Canadian Hydrographic Service, and Martin Jakobsson, Stockholm University, Sweden

6.2.1 Automatic (non-interactive) data cleaning

During this stage, the coordinates (i.e. positions and depths) obtained should be controlled automatically by a program using suitable statistical algorithms which have been documented, tested and demonstrated to produce repeatable and accurate results. When selecting an algorithm, robust estimation techniques should be taken into consideration as their adequacy has been confirmed. Many high-density bathymetry processing packages have built-in statistical processing tools for detecting and displaying outliers. Generally speaking, higher-density data sets with large amounts of overlap between lines provide an increased likelihood of detecting blunders. In addition to statistics, threshold values for survey data can be used to facilitate the detection of blunders. Each agency is responsible for the validation of the algorithm used and the procedures adopted.

All blunders and erroneous and doubtful data should be flagged for subsequent operator control. The type of flag used should indicate that it was set during the automatic stage.

6.2.2 Manual (interactive) data cleaning

Following automated processing procedures, there is a requirement for an experienced and responsible hydrographer to review the automated results and validate those results and/or resolve any remaining ambiguities.

For this stage the use of 3-D visualization tools is strongly recommended. Decision making about whether to accept or reject apparently spurious soundings can often be enhanced by viewing combined data sets in three dimensions. These tools should allow viewing the data using a zoom facility. The interactive processing system should also offer different display modes for visualization, e.g. depth plot, uncertainty plot, single profile, single beam, backscatter imagery etc. and should allow for the visualization of the survey data in conjunction with other useful information e.g. shoreline, wrecks, aids to navigation etc. Editing the data should be possible in all modes and include an audit trail. When editing sounding data, it can often be useful to understand the spatial context of the examined data points. What may appear to be bad soundings (blunders) out of context may be recognized as real seafloor artifacts (submerged piles, wrecks, etc.) when viewed in the context of a chart backdrop for example. If feasible, data displays should be geo-referenced. The ability to compare surfaces from newly collected data to ones generated from historical information can often be useful in validating the quality of the new information, or alternatively, for notifying the collecting agency of an unresolved systematic bias that requires immediate attention.

If feasible, these tools should include the reconciliation of normalized backscatter imagery with bathymetry and, provided that automated object detection tools were used, the display of flagged data for both data modes should be possible.

The rules to be observed by operators during this stage should be documented.

The flags set during the automatic stage, which correspond to depths shallower than the surrounding area, should require explicit operator action, at least, for Special Order and Order 1 a/b surveys. If the operator overrules flags set during the automatic stage, this should be documented. If a flag is set by the operator, the type of flag used should indicate this.

It can be helpful to create a map plotting ship track and date/time to assist with correcting anomalies discovered during visualization.

6.2.3 Clipping to polygon or elevation value (e.g., zero)

6.2.4 Common multibeam swath sonar errors

Spikes under nadir

Spikes may be present directly under nadir for many multibeam swath sonar systems, especially older systems. This appears to result from excessive sound energy reflected back toward the sonar and being received in the side lobes of non-nadir beams. Care should be taken when flagging these values as actual seafloor features may appear similar to these spikes.

Tracking water-column noise

Modern multibeam swath sonar systems generally include automated bottom tracking algorithms. Such algorithms search for seafloor echoes in a window centered on the previous (or adjacent) ping's depths. However, the system can begin to track water column noise, and may continue to do so if the system is not actively monitored. Depth variability between pings can be large, the across-track depth profiles are not consistent with seafloor topography, and most of the beams/pings should be discarded.

Roll-over on inward-facing steep slopes

In areas of steep slopes, especially where the seafloor rises towards the sea surface, some sonar systems have a tendency to "roll over" the outermost beams to create a backside or lip. These values must be flagged. These errors can often be detected when editing the data in the context of a priori knowledge of the seabed shape, especially man-made vertical structures or known topography of canyons or fjord walls.

Anomalies on outward-facing slopes

Where steep hills and seamounts are located off ship track, little sound energy from the sonar pulse that strikes the outward-facing slope may be reflected toward the sensor, and the quality of the returning echo is severely degraded. In such cases, the along-track distance between beams is significantly increased, and the variability in depths between pings can be large. Most of these values should be flagged. These types of errors can often be easily discovered when comparing against the overlapping deeper water swaths.

Navigation missing on first ping

In some cases, the first ping of a sonar file may be missing the associated position information supplied by the ship's onboard navigation system. This is generally a time synchronization issue, resulting from the sonar system having not yet received the position from the GPS system. This can be corrected by editing the sonar file's navigation and extrapolating the latitude and longitude backward from subsequent pings.

Inaccurate sound speed profile used

If an inaccurate sound speed profile was used in ray-tracing, the resulting across-track depth profiles tend to tilt upwards on the outermost beams on both sides ("smiles") or downwards ("frowns"). These data require the application of a better sound speed profile, which might be obtained from other, nearby sonar data, or historical averages for the region. In any event, the uncertainty in the resulting depths is expected to be large, especially on the outermost beams.

Some processing software packages have the facility to modify the observed sound speed profile in order to bend the outer beams of multiple overlapping swaths until a best fit is achieved in the overlap area.

6.2.5 Common lidar errors

Bomb craters

Most lidar data sets contain scattered too-low points, or negative blunders, perhaps produced when a specular reflection or too-close ground saturates the detector and produces an internal echo. If vegetation reflections are removed by a find-the-lowest-point-in-the-vicinity algorithm, true ground points adjacent to the negative blunders may be misidentified as vegetation reflections and removed. The result can be a conical crater that is entirely an artifact. (from Puget Sound Lidar Consortium, 2011).

Vegetation and buildings

First returns in lidar surveys commonly come from the tops of tree canopies, buildings, etc., not ground surface. The lidar data need to be classified, allowing removal of all non-bare earth returns. Some lidar systems provide backscatter intensity values that can be used to validate assumptions about types of ground cover. If geo-registered colour imagery is available, this should also be used to validate assumptions about types of ground cover.

Water-surface values

Lidar surveys (topographic) at the coast are typically flown at low tide for surveying of the inter-tidal zone. As such, water-surface values over the ocean are common, and individual waves may be identifiable. A lidar classification schema may be used to remove these values (water surface returns are generally of low intensity). Alternatively, the data may be clipped to a coastline, though clipping to a high-water coastline may eliminate valid measurements in the exposed inter-tidal zone. As a last resort, edit the data manually by paying attention to where the slope of the inter-tidal zone transitions to wave caps and troughs.

6.3 Data Validation

Contributed by Rob Hare and Paola Travaglini, Canadian Hydrographic Service, and Martin Jakobsson, Stockholm University, Sweden

The final survey data should be subject to independent in-house validation employing documented quality control procedures.

6.3.1 Internal self-consistency

comparison with other survey soundings (e.g., from overlapping swaths) on the same survey

6.3.2 External consistency

comparison with other datasets (e.g., systematic offsets), checklines from independent systems (or vessels, or dates)

6.3.3 Comparison with reported survey specifications

analysis of statistics at cross check points/lines, or in grid cells over flat seafloor, provided a sufficient number of points are available in each cell to compute reliable statistics.

References

Fugro Earthdata, 2007. Lidar Mapping Fact Sheet. http://www.fugroearthdata.com/pdfs/FCT_Lidar-Educational_11-07.pdf

International Hydrographic Bureau, 2008. *IHO Standards for Hydrographic Surveys, 5th ed., Special Pub. No. 44*, Monaco. http://www.iho-ohi.net/iho_pubs/standard/S-44_5E.pdf

Puget Sound Lidar Consortium, 2011. *Lidar artifacts*, Seattle. http://pugetsoundlidar.ess.washington.edu/lidardata/#LIDAR%20artifacts

The following has been extracted from 5th Edition, S-44 (2008) Annex B by Rob Hare and edited for context

6.4 Use of uncertainty surfaces

Many statistical bathymetry processing packages also have the ability to generate an *uncertainty surface* associated with the bathymetry using either input error estimates or by generating spatial statistics within grid cells. Displaying and codifying these *uncertainty surfaces* is one method of determining whether the entire survey area has met the required specifications. If some areas fall outside the specifications, these areas can be targeted for further data collection or use of alternative systems in order to reduce the uncertainty to within an acceptable tolerance. When performed in real-time, the sampling strategy can be adapted as the survey progresses, ensuring the collected data are of an acceptable quality for the intended use. Each agency is responsible for the validation of these processing capabilities prior to use.

6.5 Validation Procedures

The final data should be subject to independent in-house validation employing documented quality control procedures.

Chapter 7.0 Metadata Documentation

The following has been extracted from 5th Edition, S-44 (2008) Chapter 5 by Rob Hare and edited for context

7.1 Introduction

To allow a comprehensive assessment of the quality of survey data it is necessary to record or document certain information together with the survey data. Such information is important to allow exploitation of survey data by a variety of users with different requirements, especially as requirements may not be known when the survey data is collected.

7.2 Metadata

Metadata should be comprehensive but should comprise, as a minimum, information on:

- the survey in general e.g. purpose, date, area, equipment used, name of survey platform;
- the geodetic reference system used, i.e. horizontal and vertical datum including ties to a geodetic reference frame based on ITRF (e.g. WGS84) if a local datum is used;
- calibration procedures and results;
- sound speed *correction* method;
- tidal datum and reduction;
- uncertainties achieved and the respective *confidence levels*;
- any special or exceptional circumstances;
- rules and mechanisms employed for data thinning.

Metadata should preferably be an integral part of the digital survey record and conform to the "IHO S-100 Discovery Metadata Standard", when this is adopted. Prior to the adoption of S-100, ISO 19115 can be used as a model for the metadata. If this is not feasible similar information should be included in the documentation of a survey.

Agencies responsible for the survey quality should develop and document a list of *metadata* used for their survey data.

7.3 Point Data Attribution

All data should be attributed with its *uncertainty estimate* at the 95% *confidence level* for both position and, if relevant, depth. The computed or assumed scale factor applied to the standard deviation in order to determine the uncertainty at the 95% *confidence level*, and/or the assumed statistical distribution of errors should be recorded in the survey's metadata. (For example, assuming a Normal distribution for a 1D quantity, such as depth, the scale factor is 1.96 for 95% confidence. A statement such as "Uncertainties have been computed at 95% confidence assuming a standard deviation scale factor of 1.96 (1D) or 2.45 (2D), corresponding to the assumption of a Normal distribution of errors," would be adequate in the metadata.) For soundings this should preferably be done for each individual sounding; however a single uncertainty estimate may be recorded for a number of soundings or even for an area, provided the difference between the individual *uncertainty* estimates can be safely expected to be negligible. The attribution should, as a minimum, be sufficient to demonstrate that the requirements of these Standards have been met.

7.4 Bathymetric Model Attribution

If a *Bathymetric Model* is required, *metadata* should include: the model resolution; the computation method; the underlying data density; uncertainty estimate/*uncertainty surface* for the model; and a description of the underlying data.

Chapter 8.0 Gridding Data

8.1 Introduction to gridding

TEXT IN PREPARATION

8.1.1 Polynomials

TEXT IN PREPARATION

8.1.2 Near-neighbor searches

TEXT IN PREPARATION

8.1.3 Triangulation

TEXT IN PREPARATION

8.1.4 Splines

TEXT IN PREPARATION

8.1.5 Kriging

TEXT IN PREPARATION

8.2 Examples of gridding methods

8.2.1 Gridding JAMSTEC multibeam data

Contributed by K. M. Marks, NOAA Laboratory for Satellite Altimetry, USA

A grid can be formed from the individual multibeam *xyz* points downloaded from the JAMSTEC website (see Figure 5.4). Because there are so many data points it is advantageous to first take their block average using GMT routine “blockmedian,” calculating the median *z* (at the *x*, *y* location of the median *z*) for every non-empty grid cell on a 6 arc-second mesh. The next step is to use GMT routine “surface,” an adjustable tension continuous curvature surface gridding algorithm, to form a grid at 6 arc-second spacing in longitude and latitude from the median depths. Routine “grdmask” is then used to create a mask that is applied to the grid using “grdmath” so that it holds values only in cells that contain one or more of the original *xyz* points. Below we list the GMT routines used to create the grid from KR05-01 *xyz* multibeam points that is shown in Figure 8.1. We also list the GMT routines used to produce Figure 8.1.

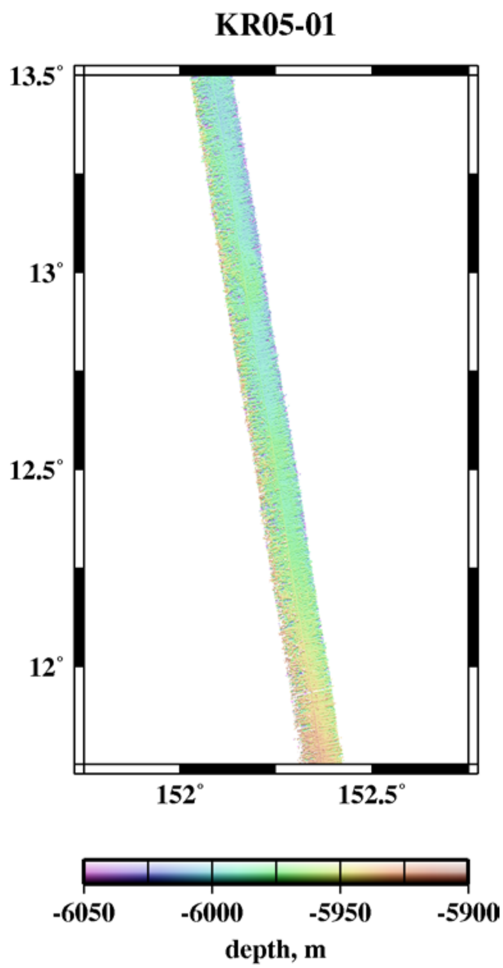


Figure 8.1 Color shaded-relief image of gridded KR05-01 multibeam points.

GMT routines that created the grid shown in Figure 8.1:

```
blockmedian KR05-01.xyz -R151.75/152.75/11.75/13.5 -I6c -Q > KR05-01.blockmedian.6c.xyz
surface KR05-01.blockmedian.6c.xyz -R151.75/152.75/11.75/13.5 -I6c -T0.25 -GKR05-01.surf.6c_grd
grdmask KR05-01.xyz -GKR05-01.mask_grd -R151.75/152.75/11.75/13.5 -I6c -NNaN/1/1 -S6c
grdmath KR05-01.surf.6c_grd KR05-01.mask_grd OR = KR05-01.surf.6c.nan_grd
```

Figure 8.1 (above) was produced with the following GMT commands:

```
grdgradient KR05-01.surf.6c.nan_grd -A0 -Ne0.2 -Ggradient_grd
grdimage KR05-01.surf.6c.nan_grd -Igradient_grd -Cmb.cpt -Jm2 -K > Fig.8.1.ps
psbasemap -R151.75/152.75/11.75/13.5 -Jm2 -Ba.5f.25:."KR05-01":WeSn -O -K >> Fig.8.1.ps
psscale -D1/-.5/2/.125h -Cmb.cpt -Ba50g25:"depth, m": -I -N300 -O >> Fig.8.1.ps
```

8.2.2 GMT Surface and Nearneighbor

We used depth values from the constrained grid cells from V12.1* in the study area as input into different gridding algorithms: GMT routine "surface," employed with the tension set to "0" and set to "1," and GMT routine "nearneighbor." Surface tension set to "0" gives the minimum curvature solution, and set to "1" gives a harmonic surface where maxima and minima are only possible at control points. "Nearneighbor" uses a nearest neighbor algorithm to assign an average value within a radius centered on a node. The GMT command lines used to produce the gridding solutions shown in Figure 8.2 are listed below.

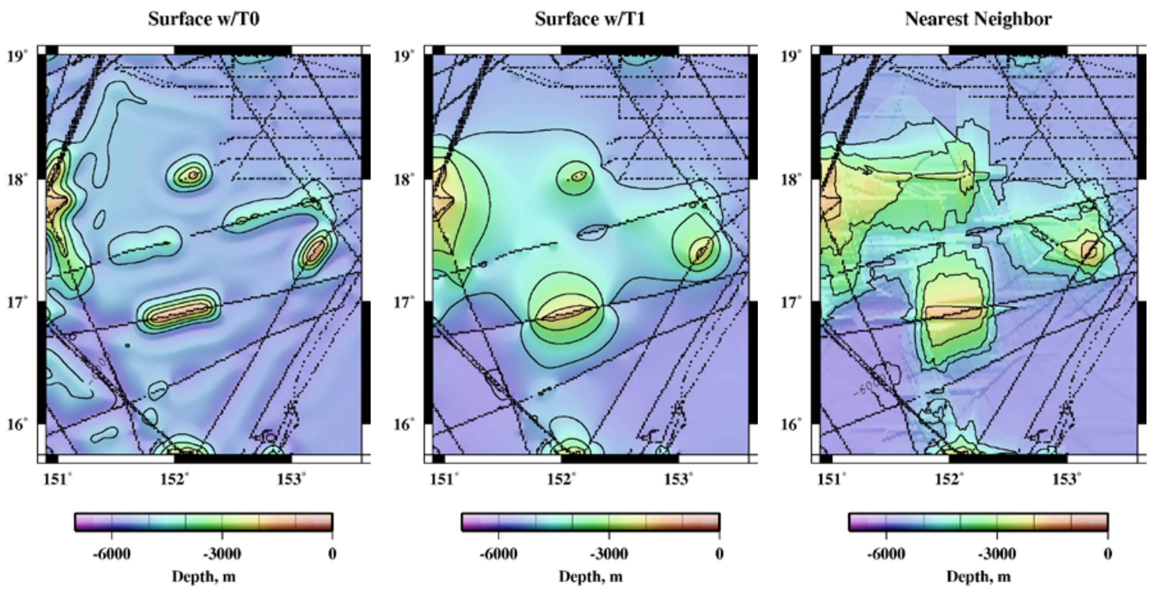


Figure 8.2 Results of gridding depths from V12.1* at constrained grid cells (black dots): GMT "surface" gridding routine with tension set to 0 (left) and set to 1 (middle), and "nearneighbor" gridding routine (right).

Gridding solutions shown in Figure 8.2 were produced with the following GMT commands:

```
img2grd -R149/154/14/19 topo_12.1*.img -Gtopo_12.1*.controls_grd -T2 -m1 -D
surface topo_12.1.nojamstec.controls.xyz -R0/2.7/0/3.41666666667 -I1m -T0 -Gsurface.t0_grd
surface topo_12.1.nojamstec.controls.xyz -R0/2.7/0/3.41666666667 -I1m -T1 -Gsurface.t1_grd
nearneighbor topo_12.1.nojamstec.controls.xyz -R0/2.7/0/3.41666666667 -N4/1 -S100k -I1m
        -Gnearneighor.1m_grd
```

8.2.3 GMT Triangulate

TEXT IN PREPARATION

8.2.4 Spline pyramid

TEXT IN PREPARATION

8.2.5 Kriging

TEXT IN PREPARATION

8.2.6 Spline Interpolation

Contributed by Chris Amante and Matt Love, NOAA National Geophysical Data Center (NGDC)

Spline interpolation estimates elevation values using a mathematical function to create a continuous surface that passes exactly through all input elevations while minimizing the overall squared curvature. Oscillation effects of a minimum curvature surface can be reduced by using tension. The surface can be thought of as a flexible rubber plate that can be adjusted using tension, with a tension of zero corresponding to the minimum curvature surface. Two open-source software programs, the Generic Mapping Tools (GMT; http://www.soest.hawaii.edu/gmt/), and MB-System (http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbsystem_home.html), offer spline gridding algorithms, ‘surface’ and ‘mbgrid,’ respectively.

Prior to using GMT’s ‘surface,’ it is recommended that the user pre-process the data with GMT ‘blockmean’, ‘blockmedian’, or ‘blockmode’ to avoid spatial aliasing and to eliminate redundant data. As an example, for two adjacent NOS surveys in xyz format, H10393, and H10394, pre-process the data using ‘blockmedian’. Required user defined parameters include the input xyz file(s), the grid spacing (-I), the region of interest (-R), and the name of the output xyz file. Numerous other options can be specified (See GMT Manual Pages).

GMT ‘blockmedian’ example:

blockmedian H10393.xyz H10394.xyz -I1c/1c -R-88.13/-88.02/30.07/30.15 > blockmedian.xyz

Required user defined parameters for GMT 'surface' include the xyz input data, the name of the output binary 2-D .grd file (-G), the grid spacing (-I), and the region of interest (-R). Numerous other options can be specified (See GMT Manual Pages). Most notably, the user can define a tension value between 0 and 1 (-T). A tension value equal to zero indicates a minimum curvature solution.

GMT 'surface' example:

surface blockmedian.xyz -Gsurface_example.grd -I1c/1c -R-88.13/-88.02/30.07/30.15 -T0.25

Required user defined parameters for MB-System's 'mbgrid' include the name of the datalist containing a list of the input data files and their format (-I), and the character string to be used as the root of the output filenames (-O). There are numerous other user-defined parameters (See MB-System Manual Pages). Most notably, the user can define the datatype (-A), the distance from the data that the spline interpolation may be applied and the interpolation mode (-C), the grid spacing (-E), the gridding algorithm mode (-F), the format of the output grid file (-G), the region of interest (-R), the tension value between 0 and infinity (-T), and the extension of the internal grid so that the output grid is a subset from the center of a larger grid allowing for data outside of the output grid to guide the spline interpolation at the edge of the output grid (-X).

MB-System 'mbgrid' example:

mbgrid -Igebco.datalist -Ombgrid_example -A2 -C10/2
-E0.00027777778/0.00027777778/degrees! -F1 -G3 -R-88.13/-88.02/30.07/30.15 -T100
-X0.05

8.2.7 Global Mapper

TEXT IN PREPARATION

8.2.8 r2v

TEXT IN PREPARATION

8.2.9 Surfer

TEXT IN PREPARATION

8.2.10 Gridding in ArcMap

Contributed by Anastasia Abramova, Geological Institute Russian Academy of Sciences, Russia

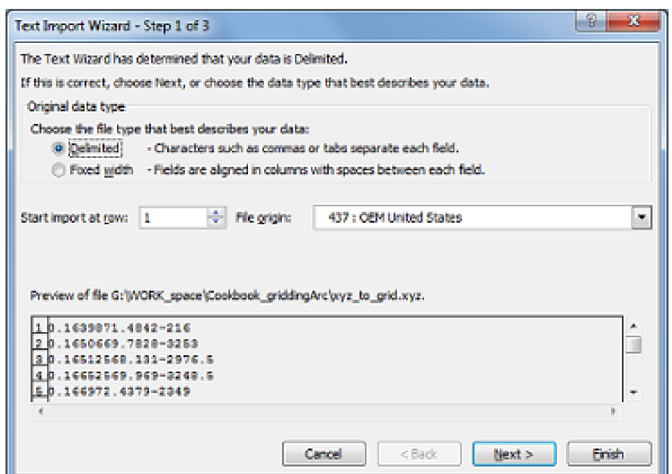
Here we present step by step example on how to open .xyz space delimited column data in ArcMap and to interpolate data points into a grid of appropriate resolution. As an example .xyz file with Latitude, Longitude and Depth in Geographic projection covering area between 2-11E and 72-74N is used. We would like to create a grid in UTM projection with appropriate cell resolution using Inverse Distance Weighting Interpolation algorithm. The procedure of reprojecting the data, assessing the data density to choose suitable grid size is described. Tools available in ArcMap in order to select optimal interpolation parameters and to compare resultant surface with original data points are discussed. In order to be able to carry out above noted you will need to have Spatial Analyst and Geostatistical Analyst extensions.

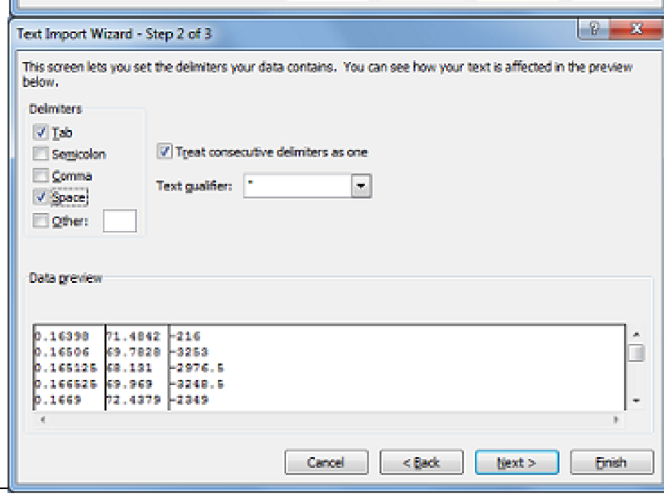
Importing .XYZ file in ArcMap

In order to open .xyz file in Arc, you need to modify it: text file needs to have header, data should be comma separated and file name should not have spaces in it.

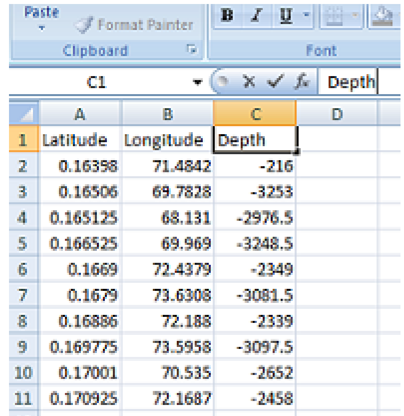
In order to modify .xyz text file, import it into MS Excel:

- File>Open .xyz
-
- Select Delimiters that you have then press Finish

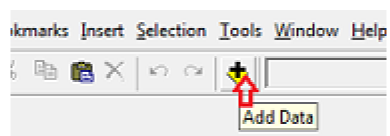




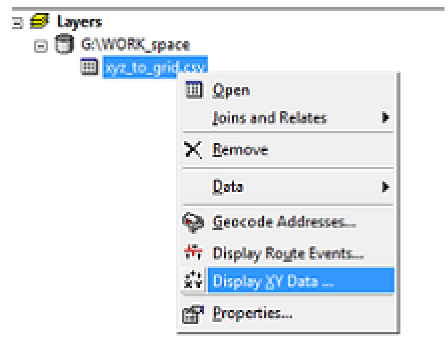
- Insert header row, type header name for each column (no spaces!)
- File> Save as .CSV (Comma delimited)
- Close file In MS Excel! (Arc doesn't want to open it if it is opened somewhere else)



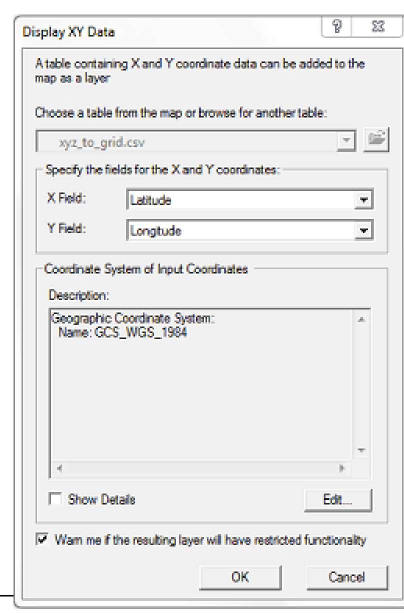
- Open ArcMap: File> New > Blank Document
- Select Add Data from Standard Toolbar (Or go to File> Add Data) and add your .csv file



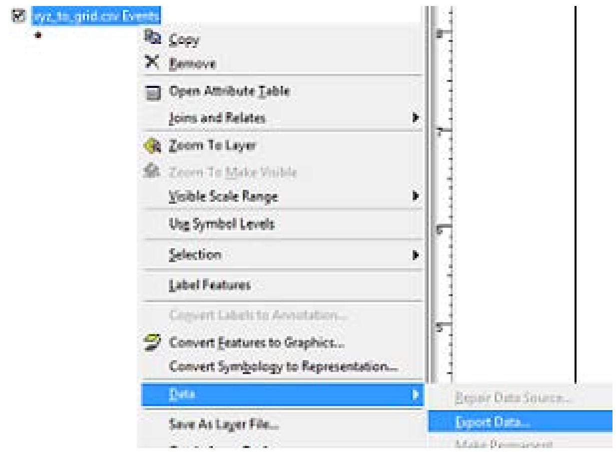
- To display your data right click on the layer> Display XY Data



- Select the appropriate fields: X as Latitude and Y as Longitude. In order to define the input data projection: press Edit > Select> Geographic> World > WGS 1984. Press Ok



- In order to save file as ArcMap shape file, right click on the layer> Data> Export Data> Save as .shp file

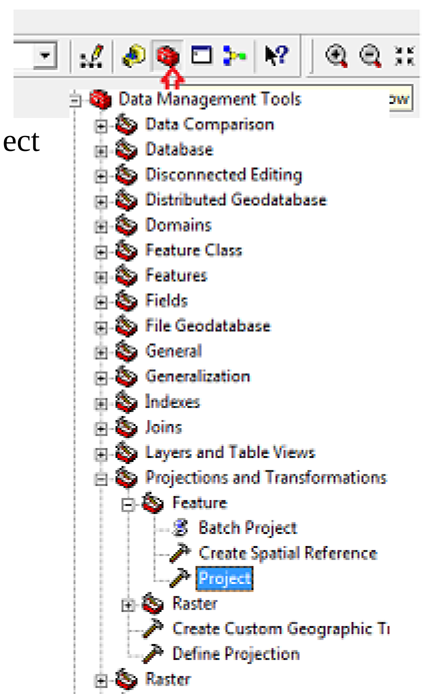


Reprojecting the data

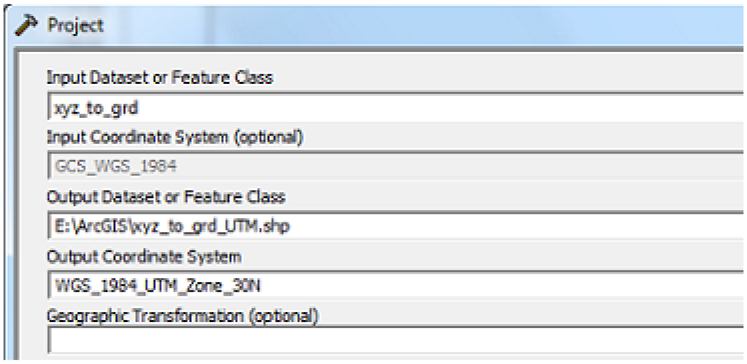
- We would like to create a grid in UTM projection with units in meters, instead of degrees. For that we need to reproject our data.

- Click on Show/Hide ArcToolbox:

- In opened ArcToolbox choose Data Management Tools> Projections and Transformations> Feature> Project

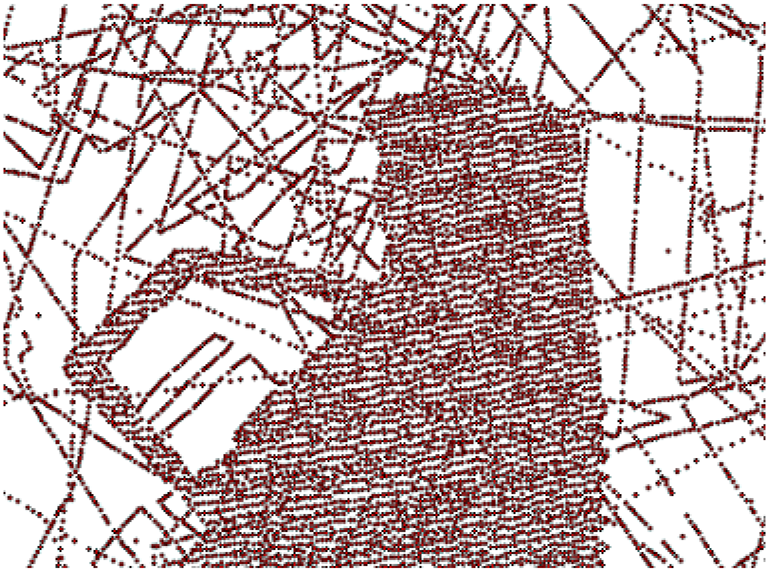


- Define the Input and Output files and Output Coordinate System (In our case it is WGS 1984, UTM Zone 30N). Press Ok.



- You might want to change the data frame projection view from Geographic to UTM_Zone_30N now (projection in the data frame view is defined by projection of the first opened file). Right click on Layers> Properties> Coordinate Systems. Select from Predefined list Projected> UTM> WGS 1984> UTM Zone 30N. Press Ok. Now data frame view has the same projection as reprojected file.

- Finally our added data looks like this. As you can notice, the data point density varies over the area



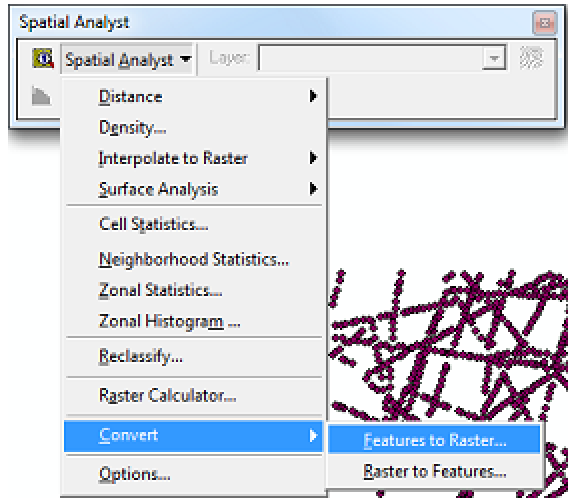
Defining cell resolution for the grid

- The choice of grid resolution depends on your purposes. Meanwhile, data distribution, density, complexity of modeled terrain and scale of the final map should be taken into account as well.

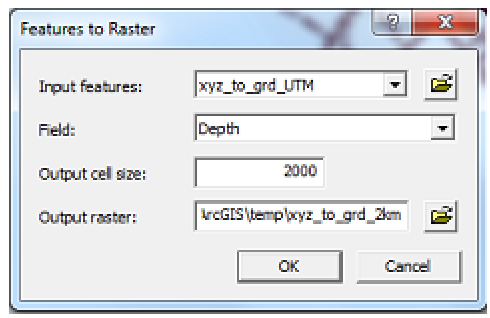
- We would like to create grid with 2 km cell resolution. Here we show how to identify the percentage of cells which values will be defined by interpolation and how to estimate the data density within each cell for the chosen grid size.

- Make sure that Spatial Analyst Extension is activated: go to Tools> Extensions> check Spatial Analyst. Also, add Spatial Analyst toolbar: go to View> Toolbars> check Spatial Analyst.

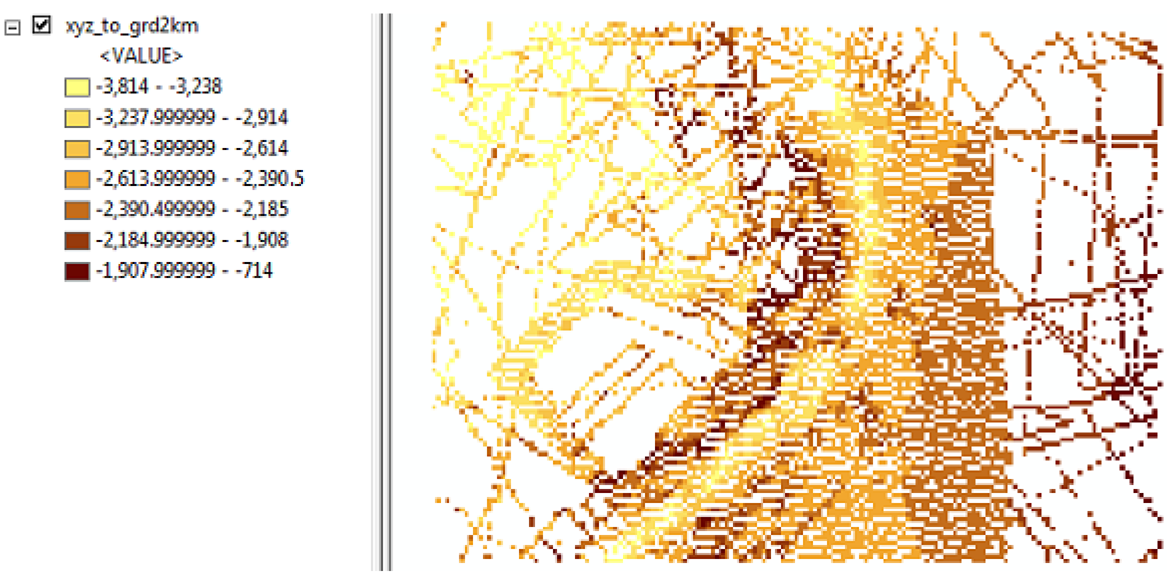
- From Spatial Analyst toolbar select Convert> Features to Raster



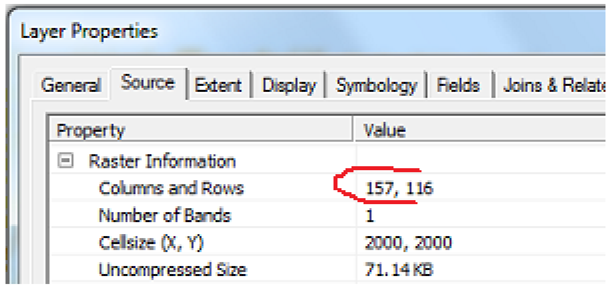
- Select appropriate field and cell resolution:



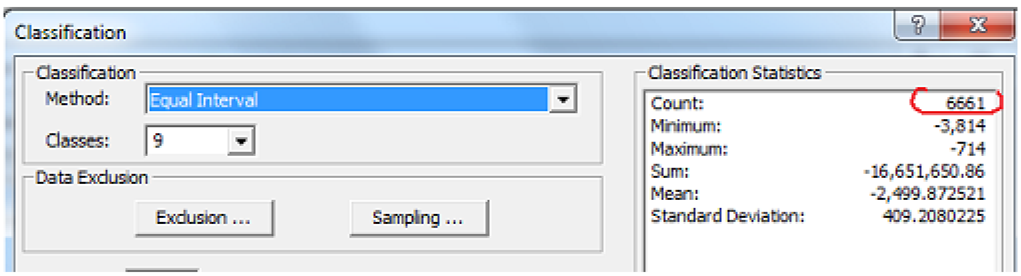
- As you can see, in the created raster, only cells with data points in it have non zero values:



- In order to find out what is the percentage of cells with “no data values”, go to Layer> Properties> Source. Here you can look up the total number of cells in the grid created by multiplying total number of rows by number of columns (157*116=18212)

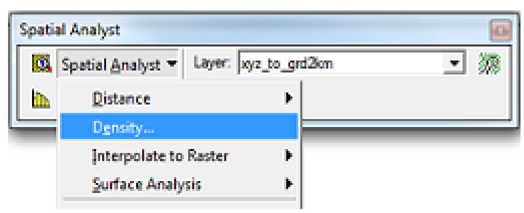


- In order to find out how many cells have non zero values, go to Symbology tab in Layer Properties window and click on Classified. Select Classify tab. The classification table will appear. In Classification Statistics Window, Count will tell the amount of non zero cells (6661). This is approximately 36 % of total amount of cells. Therefore, the values for 64% of cells will have to be estimated by interpolation.

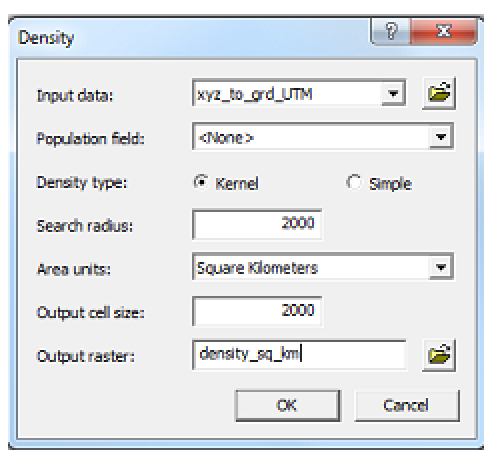


- Now we would like to find out what is the maximum density of data points per cell for the chosen grid size (we don’t want it to be too small or too big - although in the end in depends on your purpose).

- From Spatial Analyst toolbar select Density:



- Leave population field to “None”, define search radius and output cells size in projection units (meters), and desired output area units (square km)
For more information on how the density tools work go to:
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_Density_tools/009z0000000r000000/



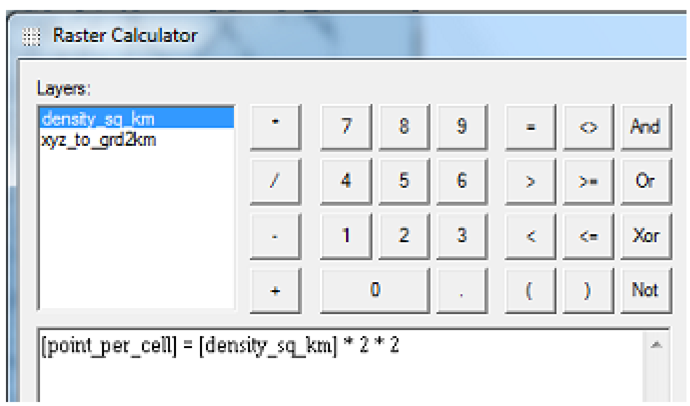
- Calculated result represents density of data points per unit area (squared km) within the neighborhood of defined search radius (2000 by 2000 square meters).

- In order to calculate amount of data points per grid cell of chosen resolution, go to Spatial Analyst toolbar and select Raster Calculator.

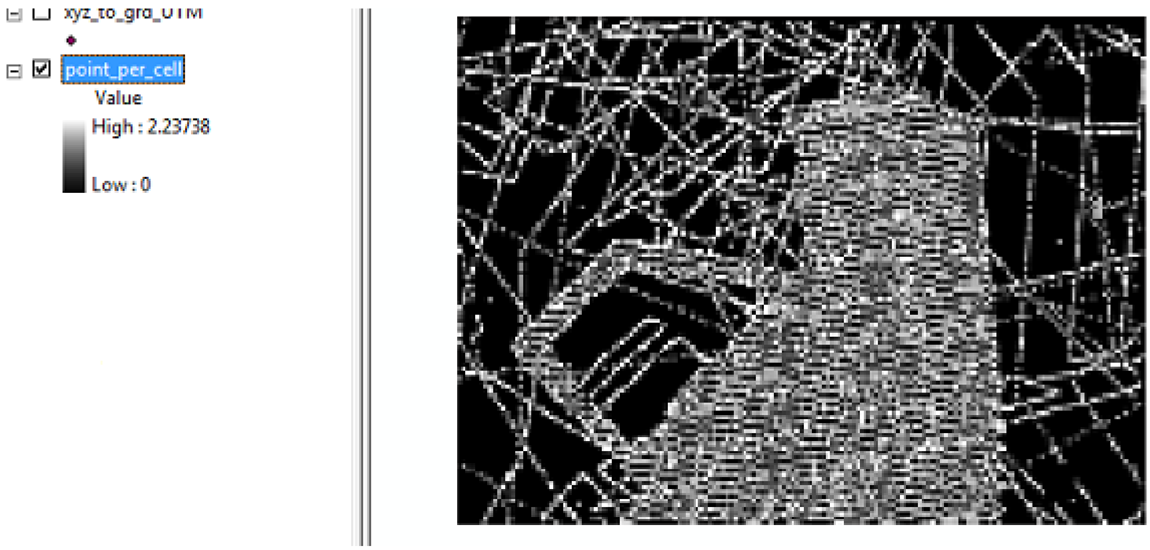
- In Raster Calculator syntax is very important. For more information on map algebra rules and on how to use raster calculator:

  http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/An_overview_of_the_rules_for_Map_Algebra/00p600000006000000/
  http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//009z000000z7000000.htm

- We would like to create a new raster layer with the name [point_per_pixel] (put layer name into square brackets and better avoid spaces in the names). We would calculate number of points per cell from the density layer by multiplying it by the area of search radius (2000 by 2000 square meters). Note that units should be the same as units in the density layer: density map was in km squared, therefore we multiply density layer by 2*2 km sq:



- As a result we get the following raster:



- As you can see, density of points is maximum 2.2 points per grid cell. This can be considered not enough; therefore you might want to change grid size to coarser resolution. We will use this grid spacing in our example.

Creating a gridded surface

- There are several interpolation algorithms available in ArcMap, these include Global and Local Polynomials, Kriging, Spline, Natural Neighbor and Inverse Distance Weighting available within Spatial Analyst and 3D extensions . The description of these interpolation algorithms can be found in other sections of Chapter 8.2. More information on interpolation from point to surface methods can be found:

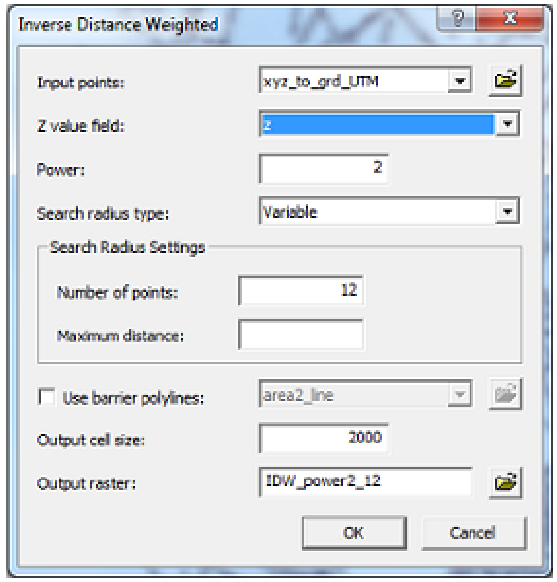
- http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_the_Raster_Interpolation_toolset
  http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_the_Interpolation_toolset

- Here we will present how to create a gridded surface using Inverse Distance Weighting (IDW) algorithm. IDW can be a good choice for fast interpolation of sparse data. IDW interpolation predicts values at unmeasured locations according to the values from the surrounding data points. Points which are closer to the prediction location have more influence (weight) on prediction than those which are further away. For more information on IDW see:

- http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=3304&pid=3302&topicname=How_Inverse_Distance_Weighted_%28IDW%29_interpolation_works

- IDW is available through ArcToolbox Spatial Analyst Tools> Interpolation>IDW (or through Spatial Analyst toolbar> Interpolate To Raster> Inverse Distance Weighting). The IDW window will appear.

- As you can see several parameters need to be defined. *Power* defines the distance weight given to each data point in the interpolation process. The higher the power, the faster the weights decrease with distance. *The number of points* defines number of points with known values to be used for prediction of the unknown value.



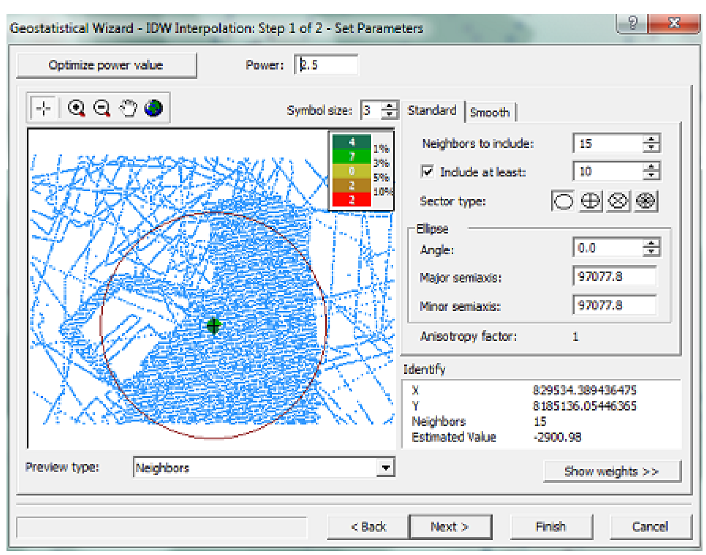
- In order to choose optimal parameters for interpolation, you can use Geostatistical Analyst toolbox. Activate the extension through Tools> Extensions> check Geostatistical Analyst. Also add it to your display through View> Toolbars> Geostatistical Analyst.

- Geostatistical Analyst> Geostatistical Wizard provides cross-validation of your model according to parameters chosen. The wizard allows you to manipulate parameters and to look at the output statistics for the created model. For more information see:

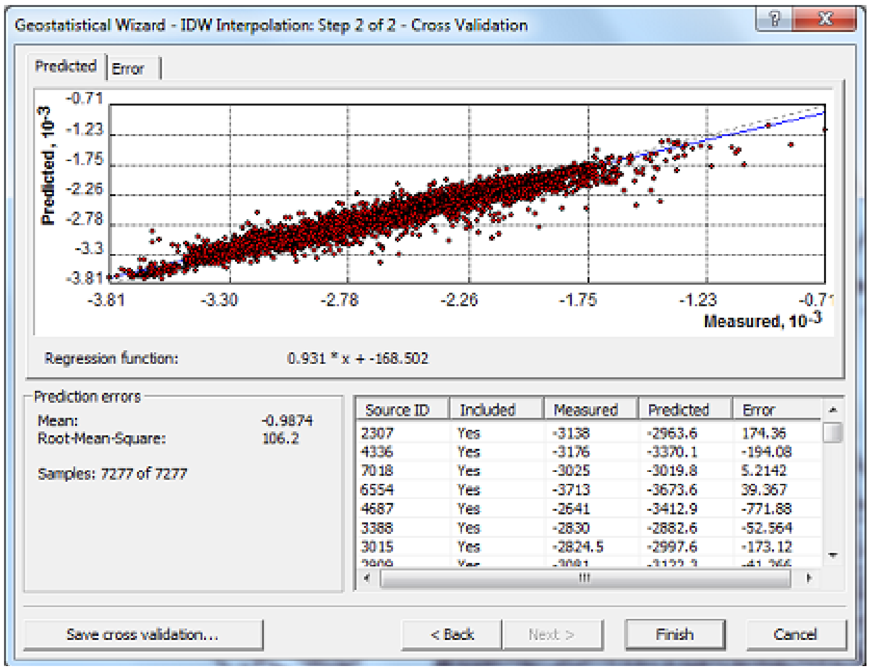
http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=3355&pid=3334&topicname=Performing_cross-validation_and_validation

- In Geostatistical Analyst open Geostatistical Wizard. Select input data point layer and attribute (Depth)

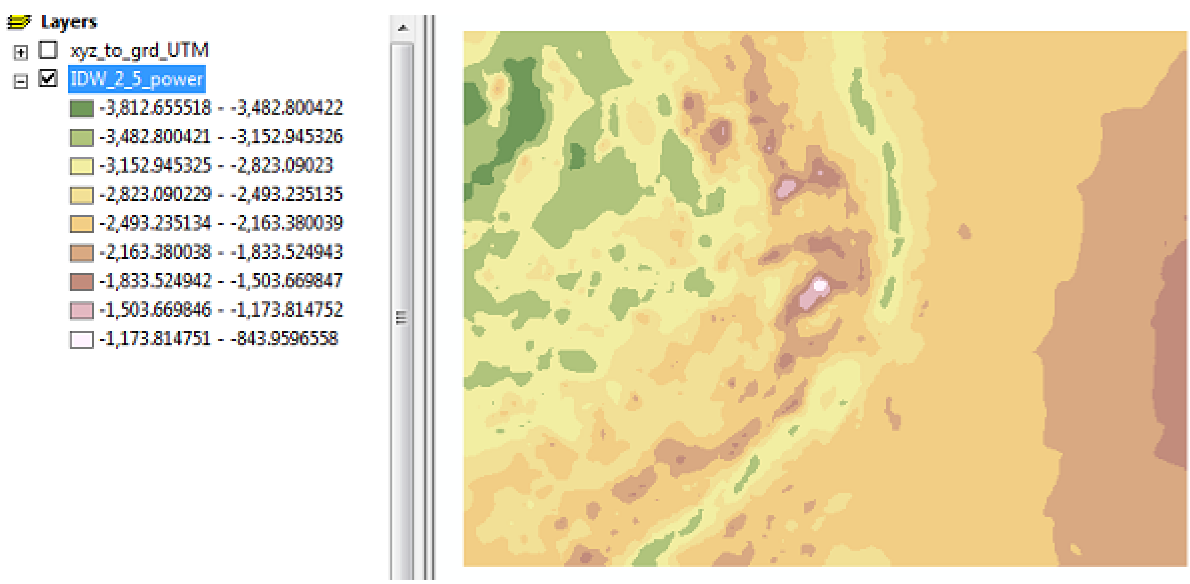
- As an example we will use Power 2.5, and minimum number of points 10. Click next.



- You can see from the figure below results of our cross-validation. The root mean square error is more than 100 meters. This can be explained by low density of data points (see figure above) and overall uncertainty of depth in the area (average depth of 2000m, with uncertainty of around … water depth). You can vary interpolation parameters and see how they affect on the model fit.



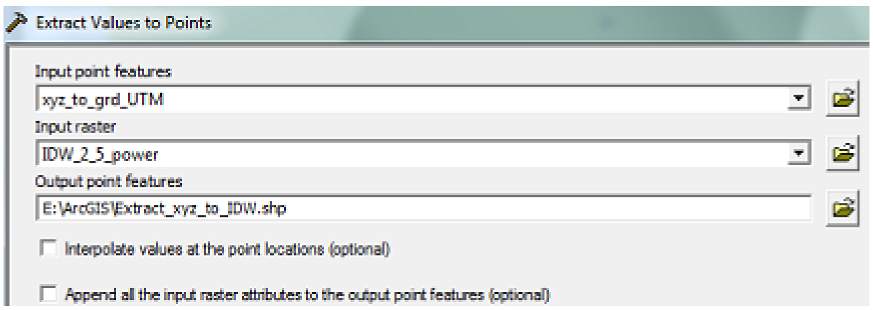
- Our final surface using above described parameters look like that:



Comparison with original data values

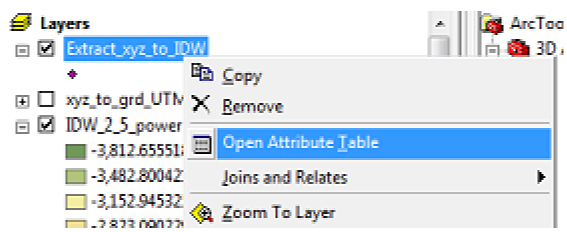
- Here we present how to compare the original data point values with created surface. This could be useful in order to investigate spatial distribution of misfit between model and the measured points. For more information on quality assessment of grids see section 9.2.1.

- You can extract values from the gridded surface corresponding to original data points with Spatial Analyst Tools> Extract Values to Points. Where Input point features are your original .xyz data points and Input raster is your interpolated raster :
-

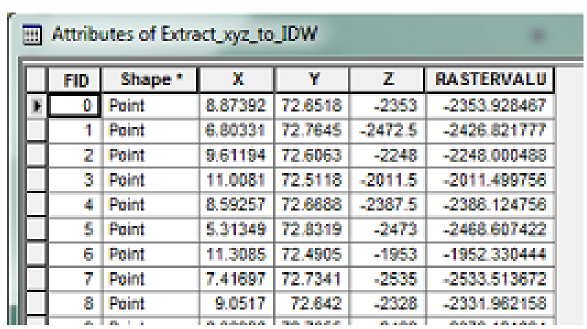


- As a result, a point feature layer is created (we call it Extract_xyz_to_IDW).

- You can view the Attribute table of the layer by right click on the layer> Open attribute table:



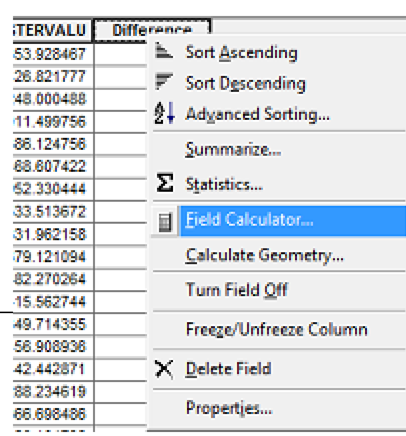
- As you can see from the table it contains information for original data points (x,y,z) and corresponding depth values from our gridded surface (RASTERVALU) for each original data point:



Attributes of Extract_xyz_to_IDW

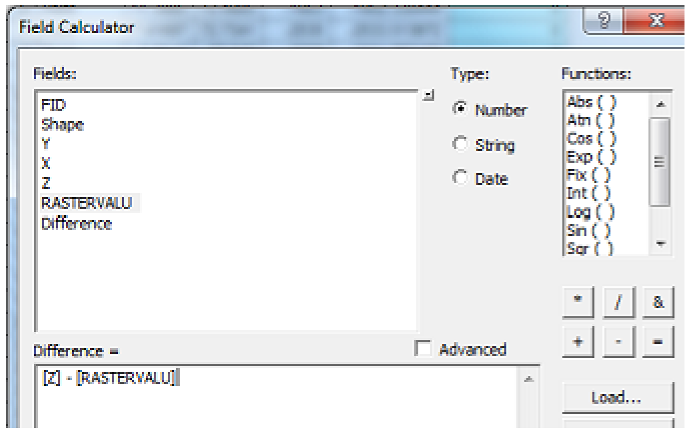
FID	Shape *	X	Y	Z	RASTERVALU
0	Point	8.87392	72.6518	-2353	-2353.928467
1	Point	6.80331	72.7645	-2472.5	-2426.821777
2	Point	9.61194	72.6063	-2248	-2248.000488
3	Point	11.0081	72.5118	-2011.5	-2011.499756
4	Point	8.59257	72.6688	-2387.5	-2386.124756
5	Point	5.31349	72.8319	-2473	-2468.607422
6	Point	11.3085	72.4905	-1953	-1952.330444
7	Point	7.41697	72.7341	-2535	-2533.513672
8	Point	9.0517	72.642	-2328	-2331.962158

- You can calculate differences between these values and visualize them.

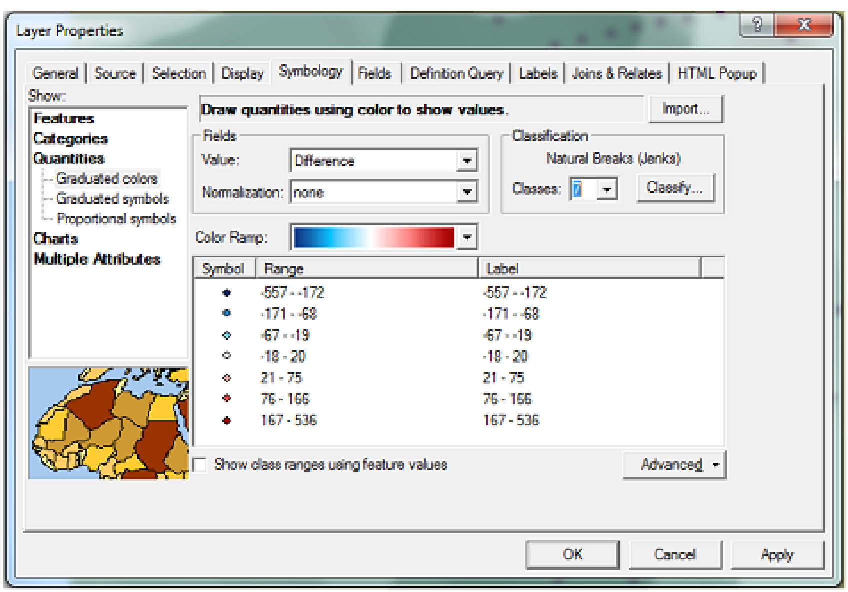


* Go to Options inside the Attribute table and select Add Field option. We will call the new column as “Difference”. After the column is created, right click on the header of new created column and select Field Calculator

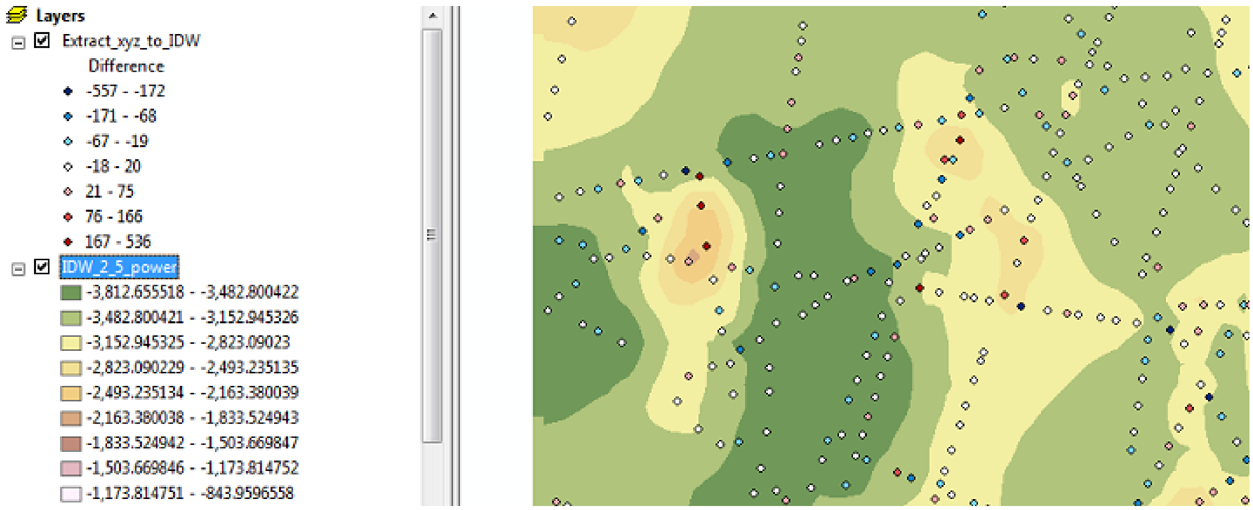
* Inside the Field calculator we can define expression for the new column values: a difference between Z and RASTERVALUE columns. After clicking Ok, the values are calculated. Close the attribute table



* Now you can visualize the Difference by right clicking on the point layer (Extract_xyz_to_IDW)> Properties. Go to Symbology tab. Select Quantities> Graduated colors. Select the Value Fields as Difference, select the color ramp and number of classes, and click OK.classes, and click OK.



* Now you can see overlaid difference results over the created gridded surface and investigate the regions where differences seem to be significant:



- For more information on analyzing gridded surfaces that you create go to:

http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=About_analyzing_raster_data
http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=603&pid=598&topicname=Surface_creation_and_analysis

See more on interpolation error analyses in other sections 8.3.

8.2.11 Gridding the International Bathymetric Chart of the Arctic Ocean (IBCAO) Version 3.0

[1]Martin Jakobsson, [1]Benjamin Hell, [1]Rezwan Mohammad, [2]Pauline Weatherall and [3]IBCAO Compilation Team

[1] Department of Geological Sciences, Stockholm University, 106 91 Stockholm Sweden
[2] British Oceanographic Data Centre (BODC), 6 Brownlow Street, Liverpool, UK
[3] See Appendix III

Introduction

This Chapter of the IHO-IOC GEBCO Cook Book describes the process used to generate Version 3.0 of the IBCAO grid – a terrain model for the Arctic Ocean region (Jakobsson et al., 2012). The description is aimed to provide the general principles of the gridding and the provided scripts can be adapted and used to generate bathymetric grids for other geographic areas.

Creating a bathymetric grid from source data sets with extreme variations in data density, e.g. sparse ship-track soundings and multibeam grids poses specific problems. Gridding routines are required that can sub-sample high resolution source data while preserving as much detail as possible in these areas and at the same time, interpolating in areas of sparse data coverage without generating gridding artifacts.

To address these issues, the IBCAO 3.0 grid has been generated in a two-stage process. Firstly a grid is created, using all the available bathymetric data, at a larger grid cell size than required for the final grid. This step helps to address the issue of generating artifacts through interpolation in areas of sparse data coverage. This grid is then sampled to the required grid cell size creating a 'base grid'. In a final step, the high resolution data sets are added on top of the 'base grid', using the remove-restore procedure ((Hell and Jakobsson, 2011; Smith and Sandwell, 1997), in order to preserve the detail in these areas.

The gridding work is done using routines from the Generic Mapping Tools (GMT) software suite (Wessel and Smith, 1991), http://gmt.soest.hawaii.edu/. In the examples given below, the GMT routines are embedded in script files for ease of use. However, if preferred, the individual GMT routines can be run separately.

The generation of the IBCAO 3.0 grid is described in the following sections:

Background to the IBCAO

The International Bathymetric Chart of the Arctic Ocean (IBCAO) project was initiated in St Petersburg, Russia, in 1997 (Macnab and Grikurov, 1997). The main goal was to create an up-to-date bathymetric portrayal of the Arctic Ocean seafloor north of 64°N. This geographic limit was chosen because the IBCAO aimed to replace the Polar Stereographic Fifth Edition GEBCO Sheet 5.17 (Johnson et al., 1979), extending from 64°N to the North Pole.

At the time that the IBCAO was initiated, Digital Bathymetric Models (DBM) were rapidly becoming the standard bathymetric database used in Geographic Information Systems (GIS) and other mapping analysis software. For this reason, a limit at 64°N was not practical for a DBM compiled on a Polar Stereographic projection with a Cartesian xy-coordinate system, so the IBCAO was extended to a cover a rectangular area as shown in Figure 8.3.

The first IBCAO was released into the public domain in 2000 and was referred to as the "Beta version" (Jakobsson et al., 2000). This version was compiled using all the bathymetric data available at that time including soundings acquired along ship and submarine tracks, spot soundings, and digitized depth contours from published maps. Land topography was included and taken from the USGS GTOPO30 topographic model (U.S. Geological Survey, 1997), except for over Greenland where the topographic grid developed by *Ekholm* (1996) was used.

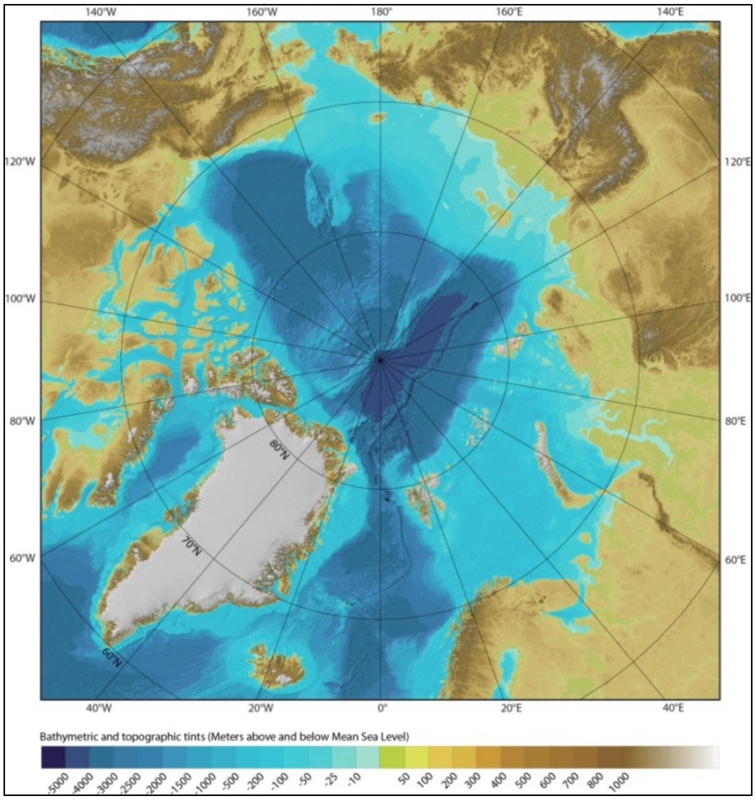


Figure 8.3 The extent of the IBCAO DBM. Projection: Polar Stereographic, true scale at 75° N, WGS 84 horizontal datum. Depths are referenced to Mean Sea Level.

The grid-resolution of the first IBCAO DBM was 2500 x 2500 m on the adopted Polar Stereographic projection. Since this first release, three major updates have been completed; Version 1.0 (Jakobsson and IBCAO Editorial Board Members, 2001), 2.0 (Jakobsson et al., 2008) and 3.0 (Jakobsson et al., 2012).

With specific emphasis on the actual gridding of the bathymetric data, this chapter explains the compilation methods used for generating the IBCAO Version 3.0, from now on referred to as IBCAO 3.0. The DBM grid resolution has been increased to 500 x 500 m and the compilations methods are improved since the Beta version from 2000.

IBCAO 3.0 Compilation

Workflow overview

IBCAO 3.0, is compiled using gridding algorithms and various data processing and filtering tools available in Generic Mapping Tools (GMT), http://gmt.soest.hawaii.edu/. GMT version

4.5.8 was used to compile IBCAO 3.0. These GMT routings are embedded in Python scripts (http://www.python.org/).

The compilation method is briefly described in the auxiliary material of Jakobsson et al. (2012). The explanation given here complements previously published method descriptions. The aim is to provide a comprehensive guide to the complete grid compilation process. The main gridding workflow of the IBCAO 3.0 is summarized in Figure 8.4. The text below explains each of the steps, A-F, in this workflow.

Compared to the previously released IBCAO 2.0 (Jakobsson et al., 2008), the gridding of Version 3.0 has been enhanced with adding source data with a spatial horizontal resolution equal to, or better than, 500 m and in a final step using the **remove-restore** method (Hell and Jakobsson, 2011; Smith and Sandwell, 1997), described in Appendix 2. This generates a grid that preserves the details to the scale of 500 m horizontally in areas with dense source data, i.e. mainly in areas covered by multibeam surveys.

In addition, a source identification grid (SID) is generated allowing users to identify the type of bathymetric source data that contributed to a particular grid cell's depth value. The generation of the source grid is not described in this chapter.

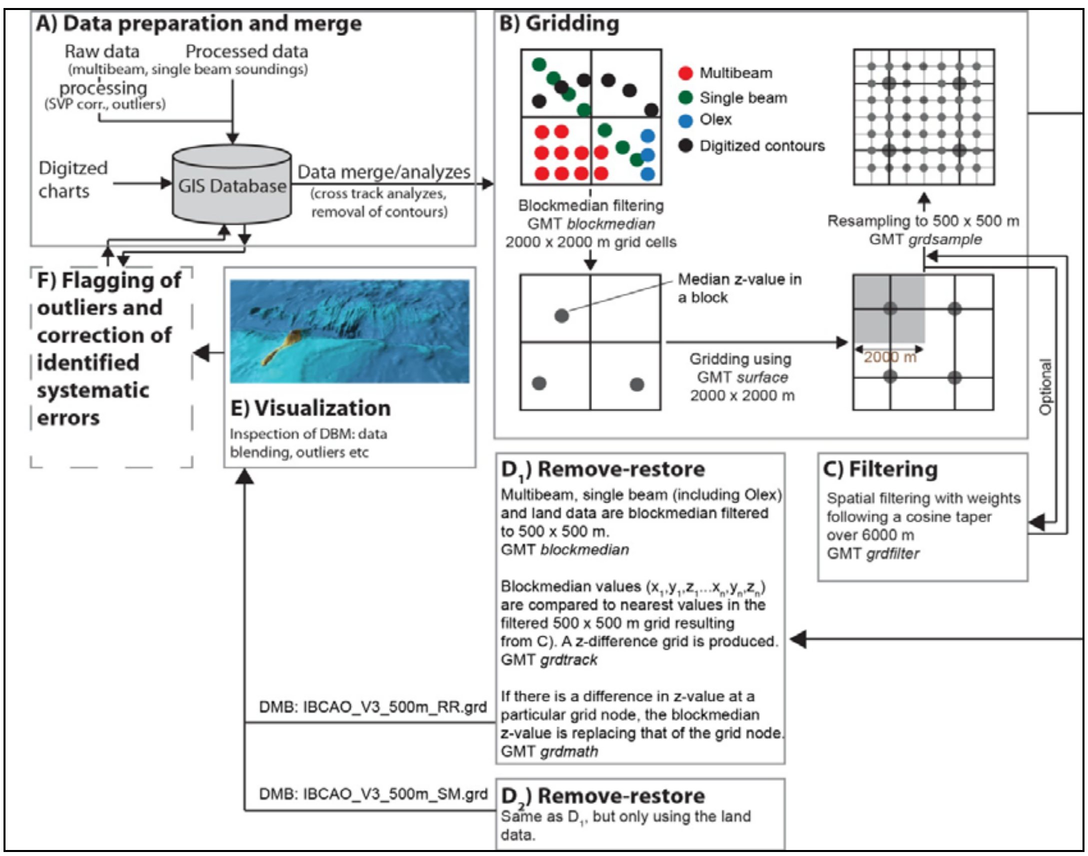


Figure 8.4 Panels A-F describes the most important components in the IBCAO 3.0 compilation process. Figure is modified from Jakobsson et al. (2012).

Data preparation and merging – workflow step A

Preprocessing of all the source bathymetric data to be used in the gridding process is done depending on the data type and origin. This will not be described further here; instead we will focus on the grid generation procedure from the point where the main preprocessing of the data has already been done.

The “clean” bathymetric source data, in the form of high resolution grids, depths soundings along ship tracks, or spot soundings are merged into a GIS database. In addition, digitized depth contours from published maps are imported as line features (polygons) into the database.

In the case of the IBCAO, Intergraph’s GIS *GeoMedia Professional* is used to visualize and analyze all the bathymetric data (Figure 8.5). It should be noted that any GIS capable of visualizing and analyzing very large amounts of point and line features with assigned z-values (depth values) and associated metadata can be used for this process.

Relevant metadata are imported along with the bathymetric data to the GIS database. For a more comprehensive discussion on metadata, the reader is referred to the relevant chapters in this Cook Book.

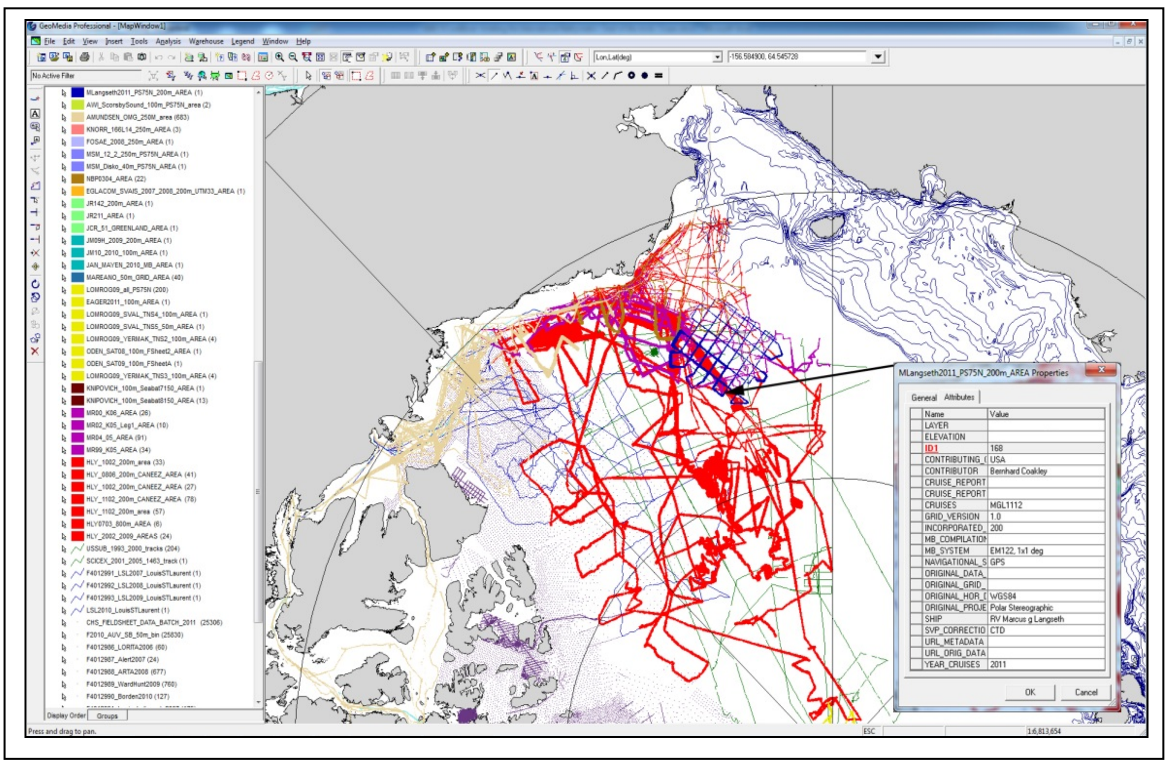


Figure 8.5 Screen view of the GeoMedia Professional GIS workspace with some of the IBCAO source data displayed. The source database resides in several databases that are accessed using GeoMedia Professional. Stored depth data with associated metadata can be displayed and analyzed. In this screen view individual soundings stored as point features are displayed as well as polygons enclosing grids contributed from multibeam surveys. The metadata of the multibeam survey with RV Marcus Langseth 2011 is shown.

The IBCAO compilation has aimed to include as a minimum the metadata listed in Table 8.1, although even this limited set of information has not been available for all bathymetric soundings used in the gridding process. The metadata may be critical for various quality assurance analyses, for example, in the estimation of random errors (Elmore et al., 2012; Jakobsson et al., 2002). The GIS database structure, including how the metadata is linked to soundings, is a comprehensive topic in itself (Hell and Jakobsson, 2008), which will not be described further here.

Although the IBCAO workflow, shown in Figure 8.4, is centered on the use of a GIS database, the actual grid compilation algorithms (B-D) operate on a suite of exported ASCII xyz flat files. This means that a grid compilation project can be setup to use the gridding procedure described here without using a GIS database as a central warehouse for bathymetric data storage.

Minimum metadata requested for contributed bathymetric grids

1. CONTRIBUTOR: (Institution, company, organization etc.)
2. CONTRIBUTING_COUNTRY: (Country codes, ISO_3166)
3. URL_ORIG_DATA: (URL to website describing the grid if available)
4. URL_METADATA: (URL to metadata by the contributor if available)
5. GRID_VERSION: (Version, could be a compilation date)
6. ORIGINAL_GRID_RESOLUTION: (Original horizontal resolution, i.e. grid cell size)
7. INCORPORATED_GRID_RESOLUTION: (Horizontal resolution of the incorporated grid)
8. ORIGINAL_PROJECTION: (Projection of the contributed grid, Mercator, UTM etc.)
9. ORIGINAL_HOR_DATUM: (Horizontal datum of the contributed grid, i.e. WGS 84, NAD1983 etc.)
10. VERTICAL_DATUM: (Vertical datum of the contributed grid, i.e. MSL, MLLW etc.)
11. ORIGINAL_DATA_SOURCES: (Multibeam/Singlebeam/Mixed etc.)

Minimum metadata requested for contributed multibeam grids

As for the contributed bathymetric grids (described above), but with the following additions:

12. MB_SYSTEM: (Brand, model and frequency)
13. SVP_CORRECTION: (CTD, XBT, uncorrected nominal sound speed etc.)
14. SHIP: (Ship name)
15. NAVIGATIONAL_SYSTEM: (RTK GPS, DGPS, GPS etc.)
16. CRUISES: (If the contributed multibeam grid is compiled from more cruises these are listed)
17. YEAR_CRUISES: (Cruise year(s))
18. PI: (Cruise PI(s))

Minimum metadata requested for contributed single beam soundings

As for multibeam grids, except for non-relevant parameters, such as grid cell size etc.

Table 8.1 Minimum metadata requested for bathymetric data contributed for inclusion in the IBCAO

Comparing the input data sets – the disambiguation process

Available GIS tools in *GeoMedia Professional* are used to compare the input data sets in a disambiguation process involving, for example, analyzes of cross-tracks and the match between digitized contours and soundings (Figure 8.6). Systematic errors are identified during this process and corrected if possible. The most commonly found systematic error is due to the sound velocity applied during the conversion from travel time to depth of echo soundings.

IBCAO has adopted the criteria that the DBM grid should contain so-called “corrected depths”, i.e. each depth should be derived from travel time using a proper harmonic mean of the water column’s sound velocity. However, several contributed bathymetric surveys were carried out using a flat nominal sound velocity of 1500 m/s, and if that not is stated in the metadata, the depths values will systematically be different from corrected depths. This inconsistency is commonly found from cross track analyzes. When such a systematic difference has been identified, a correction has been carried out using the capability of performing calculations in the GIS environment. For example, during the compilation of IBCAO 2.0 systematic differences between the included soundings from US nuclear submarine cruises were identified and adjusted (Jakobsson et al., 2008).

The digitized depth contours from bathymetric maps were used in the compilation of the first IBCAO Beta version. Since then, one of the main goals has been to remove these contours from the grid compilation and instead grid directly from the sounding database where data density permits (Figure 8.6). However, IBCAO still relies on gridding using digitized depth contours from published maps in areas of the Arctic where original soundings either are to sparse to grid or not have been made available to the IBCAO Compilation Team.

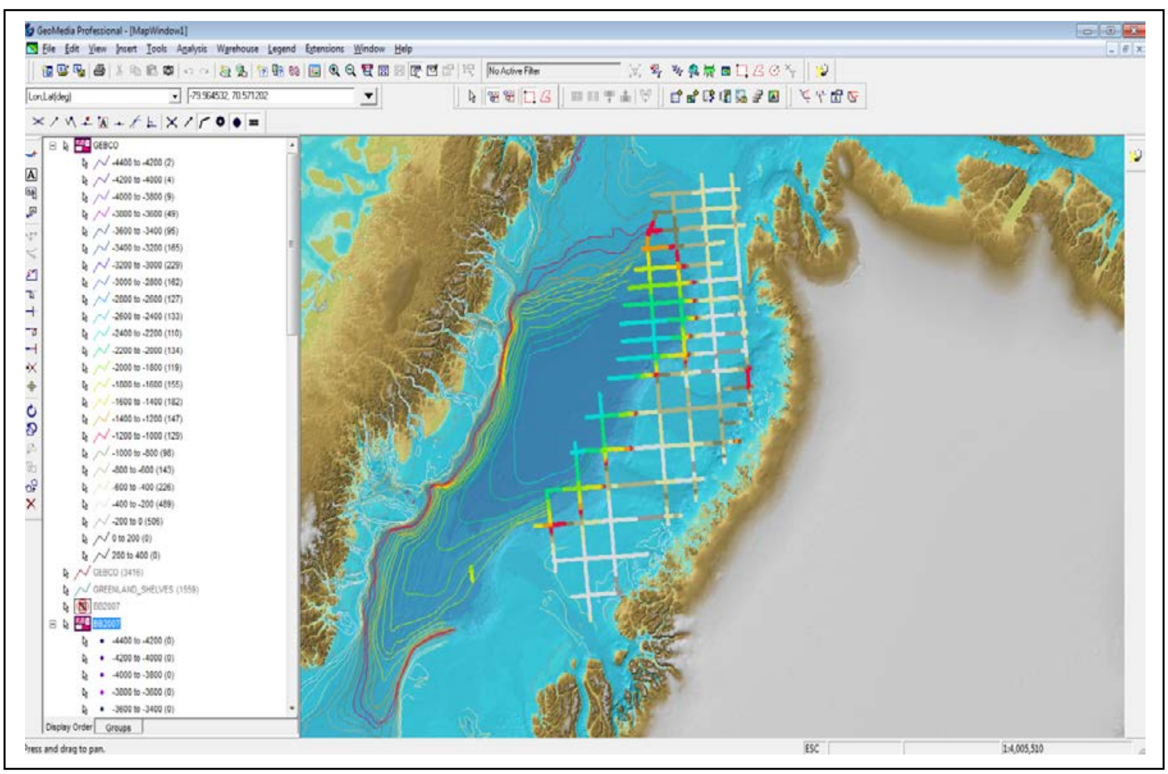


Figure 8.6 Screen view of the GeoMedia Professional GIS workspace with depth contours from GEBCO shown together with echo soundings from a seismic reflection survey contributed to IBCAO through the Geological Survey of Denmark and Greenland (GEUS). The color coding represents depths. Note how the contours are cropped at the edge of the echo sounding data. If there are noticeable mismatches between soundings and contours, edgematching may have to be carried out by moving the contours to fit the soundings. The general goal for IBCAO has been to remove contours from the compilation and instead grid directly from sounding data.

The GIS capability has also been used to carry out spatial queries to find single beam and spot soundings that reside within the boundaries of areas mapped with multibeam. When this is the case, the single beam and spot soundings are flagged and removed from the gridding process.

Generating ASCII xyz files for use in the gridding process

After the time consuming disambiguation process, all data that pass the set criteria for use in the gridding process are exported from the GIS database as xyz points. In the case of depth contours, the xyz values of their nodes are exported.

The horizontal datum of the IBCAO is WGS84 and the projection is Polar Stereographic with a true scale at 75°N. Therefore, the xy-coordinates for each exported depth (z-value) are based on this projection and datum.

Each multibeam or single beam survey, digitized contours from a bathymetric map, or collection of spot soundings is exported to separate ASCII files. This permits the generation of the SID grid and the option to use different kinds of data in the different gridding steps shown in Figure 8.4 B-D.

Gridding – Workflow steps B and C

The IBCAO gridding is done using GMT routines that are embedded in two Python scripts. Other scripting languages such as Perl or AWK can be used or you can run the GMT commands individually without embedding them in a script.

However, there are numerous advantages to using a scripting language. For example, it is possible to work with variables and the script programming language allows the user to construct their own data processing routines.

We use two Python scripts:

- One for the main gridding procedure to develop the 'base grid'
- One for the remove-restore procedure to add the higher-resolution data sets on top of the 'base grid'.

IBCAO was gridded on a computer with Windows 7 (64-bit). The Python scripts are executed in a DOS-prompt, and the GMT specific commands are executed in Python using:

```
os.system('gmtcommand')
```

Both GMT and Python are available for other operating systems, and the procedure described here is not limited to Windows.

A simplified version of the IBCAO Python gridding script is supplied in Appendix 1. The simplification consists of removing several internal data checking procedures in order to make the gridding recipe easier to follow.

Generating the ‘base grid’ – the gridding script

Explanations of the components of this script are given below and are numbered to match the corresponding sections in the script outlined in Appendix 1.

1) Setup the input variables

There are numerous variables that are defined here to be used in the Python script. The concept has been that instead of changing, for example the grid cell size, boundary box or data path in all the individual GMT routines that will use these parameters, variables are set so it is necessary to change their values only in one place. The different variables are described in Appendix 1.

2) Creation of a list of input files for use in the gridding process

A list of all the individual ASCII xyz-files that will be used in the gridding process is assembled as in the example Python script below:

```
FILElist=['file1_SB.xyz', 'file2_MB.xyz', 'file3_CONT.xyz','file4_TOPO.xyz']
```

In this example, four xyz files are listed for use in the gridding process. Further files can be added to this list.

At this stage, we do not separate the data files depending on their data type, e.g. multibeam, spot soundings, contour or grid etc. there is no separation of the different data categories as the first grid created will constitute a “base grid” developed from all the available data.

A separation is made later in the process where remove-restore is used to add all the higher resolution single beam, multibeam and topography data on top of the “base grid”.

3) Check data consistency and merge files

When large amounts of data are exported from a database, there may occasionally be some problems with data inconsistency such as, for example, an empty line, a missing coordinate, or missing depth value. Therefore a simple loop is included that goes through every line in the input files and check if three xyz values are included. This loop can be extended with additional criteria, such as maximum allowable depth or height etc.

The output from this loop is one large file of data merged from the individual input files. The name of this merged file is set by variable **DATAstr**, defined in the ‘set variables’ section of the Python script file.

4) Segment the data into regions to handle large data amounts with GMT "blockmedian"

This tiling of the data set may only be necessary when working with large volumes of data. For too large datasets, the GMT program **blockmedian** may not be able to allocate enough memory.
It was noted that when the data set is comprised of more than approximately 100 million soundings, and the spatial area to be gridded is as large as the IBCAO, then the available RAM became an issue. The exact limitation has not been investigated. To further boost the available memory the following two DOS commands have been added to the script:

```
set EMM=RAM

set dos=high, umb
```

Within the Python script, the GMT routine **gmtselect** is used to cut the data into four tiles.

The following example shows the use of **gmtselect** to extract a subset from the IBCAO source data set – generating file R1.xyz

Routine as shown in python script I:

```
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R1,DIRstr, DATAstr, DIRstr, R1))
```

Version using parameters from the 'setup variables' section of python script I:

```
gmtselect –R0/2910000/0/2910000 -V C:\\IBCAOData\\IBCAOinput.xyz > C:\\IBCAOData\\R1.xyz
```

5) Block median filtering

Block median filtering is shown as the first step in the gridding process in Figure 8.4B. This is required as a data preprocessing step before the subsequent surface spline in tension gridding is carried out using the GMT routine **surface** (Smith and Wessel, 1990).

Block median filtering provides a way of decimating the input data so that a maximum of one z-value remains for each grid cell. This step is done as it is not possible to fit a surface exactly through more than one point at each grid node; the surface is then simply over determined.

There are several ways to decimate your data, in the case of developing a bathymetric grid, one of the reasons for using the median value within every non-empty cell is that this will decrease the influence of potential outliers compared to if the mean is used.

There are a number of options available with the GMT routine **blockmedian**. For developing the, IBCAO 3.0 the option **–Q** is used. This finds the median z-value within the defined grid cell size and uses its xy coordinates. Blockmedian's default is that the median x and median y is found independently of median z.

In the case of the IBCAO 3.0, the grid cell size in the block median filtering processing was set, with the option **–I**, to 2000 m. This should be the same as the cell size that will be used when gridding the data with the GMT **surface** command.

The following example shows the use of **blockmedian** to median filter the subset of the IBCAO source data set.

Routine as shown in python script I:

```
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R1,BM_I, DIRstr, R1, DIRstr, BM_OUTPUT1))
```

Version using parameters from the 'setup variables' section of python script I

```
blockmedian -R0/2910000/0/2910000 -I2000 -V -Q C:\\IBCAOData\\R1.xyz > C:\\IBCAOData\\\OUTFILE_BM1.xyz
```

6) Merge the GMT Blockmedian filtered segments into one file

This step consists of Python loops that open each of the four files of block median values and merges them all into one file that will be used as input to the gridding process using GMT's **surface .**

7) Gridding, crop, filter and re-sampling

This section covers the gridding, filtering and re-sampling of the 'base grid', generated from all the available source data. The first step consists of gridding all the data that have been filtered with blockmedian with the GMT routine **surface**. The following key parameters are set for **surface**:

-R This sets the geographic limits of the grid and should be same as the combined subset regions created in step 4. Note that a slightly larger area is gridded than is required for the final grid. It may be enough to extend the gridded region with a few grid cells in all directions. Edge effects often occur during the gridding, and by applying a larger area here the outer edges may be cropped subsequently.

-I Grid cell size in the same units as the data, in the case of the IBCAO this is meters. The cell size should be the same as used in the **blockmedian** filtering process. IBCAO 3.0 used 2000 m.

-N Maximum number of iterations in each gridding cycle. The default is 250. Complex data may require more than 250 iterations in order to allow the surface to fully converge to the data points. For IBCAO 3.0 compilation –N5000 was used.

-T Tension of the minimum curvature surface. A value of 0 applies no tension and gives a minimum curvature smooth solution while a value of 1 applies maximum tension, which gives a harmonic spline solution where local max/min z-values only occur at real data points. Trial and error has resulted in the IBCAO grids being compiled using values between 0.32 and 0.4.

-Lu The parameter has recently been added to the GMT surface routine. It allows the user to constrain the range of the output grid, in this case, setting an upper limit of the z range of the grid. *For the IBCAO V3.0, this was set to -0.1*. IBCAO is gridded with both topography and bathymetry data in order to get a good representation of the coastline. However, the grid created in this first step is limited to only contain values below the coastline. The topography is later merged onto this grid using the remove-restore concept.

An example of the use of the GMT **surface** routine for gridding the blockmedian filtered data:

Routine as shown in python script I:

```
os.system('surface %s\\%s -R%s -I%s -N%s -T%s -G%s\\%s -Lu%s -V' % (DIRstr, BM_OUTPUT, BM_R, S_I, N_I,
S_T, DIRstr, S_grd_out, LU))
```

Version using parameters from the ‘setup variables’ section of python script I:

```
surface C:\\IBCAOData\\OUTFILE_BM.xyz -R-2910000/2910000/-2910000/2910000 –I2000
–N5000 –T0.32 –G C:\\IBCAOData\\out_grd.grd –Lu-0.1 –V
```

The grid generated with the GMT **surface** command above is cropped to its final extent using **grdcut** and filtered using **grdfilter**. The latter is optional, and has been implement specifically as all the high resolution data will again be merged onto this “base grid” using remove-restore.

An example of using GMT’s **grdcut** to crop the grid generated using **surface**:

Routine as shown in python script I:

```
grdcut %s\\%s -R%s -G%s\\%s' % (DIRstr, S_grd_out, CUT_R, DIRstr, CUT_grd_out
```

Version using parameters from the ‘setup variables’ section of python script I:

```
grdcut C:\\IBCAOData\\out_grd.grd -R-2904000/2904000/-2904000/2904000 -G C:\\IBCAOData\\out_grd01.grd
```

There are several types of spatial filters to choose between. For IBCAO 3.0 a cosine arch was selected with –Fc6000. This means that a weighted averaging with cosine arc weights are applied over 6000 m. The parameter –D0 means that the set filter width is in the same unit as the grid.

An example of using GMT's **grdfilter** to filter the cropped grid:

Routine as shown in python script I:

```
'grdfilter %s\\%s -G%s\\%s -R%s -D0 -Fc%s' % (DIRstr, CUT_grd_out, DIRstr, FILT_grd_out, CUT_R, FILT))
```

Version using parameters from the 'setup variables' section of python script I:

```
grdfilter C:\\IBCAOData\\out_grd01.grd -G C:\\IBCAOData\\filt_grd01.grd -R-2904000/2904000/-2904000/2904000 -D0 –F6000
```

The final step of the gridding procedure, shown in Figure 8.4B, consists of resampling the grid to a higher resolution using GMT's **grdsample**. This resolution should match the desired resolution of your final grid. In the case of IBCAO 3.0 the grid is resampled to 500x500 m.

An example of using GMT's **grdsample** to sample the cropped/filtered grid to the desired resolution of your final grid:

Routine as shown in python script I:

```
os.system('grdsample %s\\%s -G%s\\%s -R%s -I%s -V' % (DIRstr, FILT_grd_out, DIRstr, RESAMP_grd_out, CUT_R, R_I))
```

Version using parameters from the 'setup variables' section of python script I:

```
grdsample C:\\IBCAOData\\filt_grd01.grd -GC:\\IBCAOData\\resamp_grd01.grd -R-2904000/2904000/-2904000/2904000 –I500 -V
```

This re-sampled grid is now the '**base grid**' to be used as a base for in the remove-restore process described below.

Remove-restore – Workflow step D

The gridding steps and algorithms used to develop the IBCAO 2.0 are nearly identical to those described above. However, for the generation of the IBCAO 3.0, a final of step of adding the multibeam, dense single beam echo soundings and land topography on top of the 'base grid' using the **remove-restore** method was included (Hell and Jakobsson, 2011; Smith and Sandwell, 1997).

This final step, generates a grid that preserves the detail to the scale of 500 m horizontally in areas with dense source data, i.e. mainly in areas covered by multibeam surveys. The value of the predicted bathymetry is added back to the interpolated difference.

The **remove-restore** procedure uses a set of GMT routines. The first step is to calculate the difference between two gridded data sets. These difference values are then gridded and added on top of the base grid. This preserves the resolution of the input data.

For the IBCAO V3.0 grid generation work, dense single beam echo soundings refers to a spatial coverage where the soundings are closer to or less than 500 m apart. This is because the target resolution of the final grid is 500 x 500 m. Sparse random tracklines are omitted from the remove-restore process.

The grid used in this step is the 500 x 500 m 'base grid' created from re-sampling of the filtered 2000 x 2000 m grid, Box C in Figure 8.4.

The following gives a description of the components of the **remove-restore** python script in Appendix 2. The description is organized so that each numbered explanation is marked with a corresponding number in the script. The numbering starts from 8 in order to avoid confusion with the description of the script in Appendix 1.

8) Setup the input variables

As described in section 1 above, the variables to be used in the GMT routines and Python script are setup. These variables are defined in the script in Appendix 2.

9) Merge all the input data

The separate files of multibeam and dense single beam echo soundings and land topography data are merged into one file. In the case of the IBCAO 3.0, this also includes the Olex echo sounding data collected primarily by fishing vessels (see Jakobsson et al., 2012).

The merge is done by reading all the input files and checking that there are three variable provided (this procedure is described in section 2 above). One merged output file is written containing all the data that will be added onto the input 'base grid' grid using remove-restore.

10) Cut the data set into sub-regions

This step may only be necessary when working with large volumes of data which may cause memory issues when using the GMT routine **blockmedian**. See explanation 4 above.

In this example, the outputs from this segmentation process are the files R1_RR.xyz…… R4_RR.xyz. Note that the number and geographic division of the regions can be modified to fit the data volume and spatial distribution.

11) Block median filtering

Block median filtering provides a way of decimating the input data so that a maximum of one z-value remains for each grid cell. This step is done as it is not possible to fit a surface exactly through more than one point at each grid node; the surface is then simply over determined. See explanation 5 above.

Routine as shown in python script II:

```
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R1,BM_I, DIRstr, R1, DIRstr,
BM_RR_OUTPUT1))
```

Version using parameters from the 'setup variables' section of python script II:

```
blockmedian -R'0/2910000/0/2910000' -I500 -V -Q m:\my_data\R1_RR.xyz >
C:\\IBCAOData\\OUTFILE_RR_BM_1.xyz
```

12) Merge the blockmedian filtered segments

One output file is created from this process. The name of the file is set by variable IBCAO_RR_SEGMENT. For further details, see explanation 6 above.

13) Find the depth in the 'base grid' at the location of the blockmedian filtered data points

This next step finds the depth in the 'base grid' at the location of the blockmedian filtered data points. This is done so we can compare the difference in z-values between the high resolution data that were blockmedian filtered and the resampled 500 x 500 m base grid.

The following line in the script will take the merged blockmedian filtered data, the resampled 500 x 500 m grid and interpolate from the grid at the points in the input file.

Routine as shown in python script II:

```
os.system('grdtrack %s -G%s -V -S > %s' % (IBCAO_RR_SEGMENT,IBCAO_resamp,IBCAO_resamp_predict))
```

Version using parameters from the 'setup variables' section of python script II:

```
grdtrack IBCAO_RR_SEGMENT.xyz –Gresamp_grd01.grd -V -S > resamp_grd01_pred.xyzz
```

The resulting output ASCII file, "resamp_grd01_pred.xyzz" , looks like:

x_1	y_1	z_{1bm}	z_{1i}
..	..	..	..
x_n	y_n	z_{nbm}	z_{ni}

where x_1,y_1,are the coordinates for a blockmedian depth z_{1bm} created in step 11) and z_{1i} the depth retrieved from the 500 x 500 m grid at this location.

14) Compute the difference in z-values between the blockmedian filtered data and the base grid That is the difference between z_{1bm} and z_{1i} in the example above. If the depth difference here is bigger than a threshold set in the script detailed in Appendix 2 (variable DIFF_P) it is written to a file: IBCAO_diff.xyz

15) Create a difference grid

A grid is created, using the GMT nearneighbor routine, that consist of the differences between the blockmedian filtered data and the resampled 500 x 500 m base grid. The neighbor routines reads x,y,z from an input file and uses a nearest neighbor algorithm to assign an average value to each node that have one or more points within a radius centered on the node.

An example is given below of gridding with nearneighbor:
Routine as shown in python script II:

```
os.system('nearneighbor %s -G%s -R%s -I%s -S1000 -N4 -E0 -V' %
(IBCAO_resamp_diff,IBCAO_resamp_diff_GRD,BM_R,BM_I))
```

Version using parameters from the ‘setup variables’ section of python script II:

```
nearneighbor IBCAO_diff.xyz –GIBCAO_diff.grd –R-2904000/2904000/-2904000/2904000 –I500 -S1000 -N4 -E0 -V
```

16) Merge the difference grid on top of the base grid

The difference grid, from step 15, above is ‘added’ to the base grid using the GMT routine **grdmath** to create the final grid. In the example given below, the final grid file is called IBCAO_final_RR.grd.

Routine as shown in python script II:

```
os.system('grdmath %s %s ADD = %s' % (IBCAO_resamp,IBCAO_resamp_diff_GRD,IBCAO_final))
```

Version using parameters from the ‘setup variables’ section of python script II:

```
grdmath resamp_grd01.grd IBCAO_diff.grd ADD = IBCAO_final_RR.grd
```

References

Ekholm, S., 1996, A full coverage, high-resolution, topographic model of Greenland computed from a variety of digital elevation data: Journal of Geophysical Research, v. 101, no. B10, p. 21961-21972

Elmore, P. A., Fabre, D. H., Sawyer, R. T., and Ladner, R. W., 2012, Uncertainty estimation for databased bathymetry using a Bayesian network approach: Geochem. Geophys. Geosyst., v. 13, p. Q09011.

Hell, B., and Jakobsson, M., 2008, Data Model and Processing Environment for Compiling Heterogeneous Bathymetric Data: International Hydrographic Review, v. 9, no. 1, p. 23-33.

Hell, B., and Jakobsson, M., 2011, Gridding heterogeneous bathymetric data sets with stacked continuous curvature splines in tension: Marine Geophysical Research, v. 32, no. 4, p. 493-501.

Jakobsson, M., Calder, B., and Mayer, L., 2002, On the effect of random errors in gridded bathymetric compilations: Journal of geophysical research, v. 107, no. B12, 2358, p. 1 - 11.

Jakobsson, M., Cherkis, N., Woodward, J., Macnab, R., and Coakley, B., 2000, New grid of Arctic bathymetry aids scientists and mapmakers: EOS, Transactions American Geophysical Union, v. 81, p. 89, 93, 96.

Jakobsson, M., and IBCAO Editorial Board Members, 2001, Improvement to the International Bathymetric Chart of the Arctic Ocean (IBCAO): Updating the Data Base and the Grid Model, *in* Union, A. G., ed., American Geophysical Union Fall Meeting, Volume 82: San Francisco, CA, USA, EOS Transactions, p. OS11B-0371.

Jakobsson, M., Macnab, R., Mayer, L., Anderson, R., Edwards, M., Hatzky, J., Schenke, H. W., and Johnson, P., 2008, An improved bathymetric portrayal of the Arctic Ocean: Implications for ocean modeling and geological, geophysical and oceanographic analyses: Geophysical Research Letters, v. 35, p. L07602.

Jakobsson, M., Mayer, L., Coakley, B., Dowdeswell, J. A., Forbes, S., Fridman, B., Hodnesdal, H., Noormets, R., Pedersen, R., Rebesco, M., Schenke, H. W., Zarayskaya, Y., Accettella, D., Armstrong, A., Anderson, R. M., Bienhoff, P., Camerlenghi, A., Church, I., Edwards, M., Gardner, J. V., Hall, J. K., Hell, B., Hestvik, O., Kristoffersen, Y., Marcussen, C., Mohammad, R., Mosher, D., Nghiem, S. V., Pedrosa, M. T., Travaglini, P. G., and Weatherall, P., 2012, The International Bathymetric Chart of the Arctic Ocean (IBCAO) Version 3.0: Geophys. Res. Lett., v. 39, no. 12, p. L12609.

Macnab, R., and Grikurov, G., 1997, Report: Arctic Bathymetry Workshop: Institute for Geology and Mineral Resources of the Ocean (VNIIOkeangeologia).

Smith, W. H. F., and Sandwell, D. T., 1997, Global seafloor topography from satellite altimetry and ship depth soundings: Science, v. 277, p. 1957-1962.

Smith, W. H. F., and Wessel, P., 1990, Gridding with continuous curvature splines in tension: Geophysics, v. 55, p. 293-305.

U.S. Geological Survey, 1997, GTOPO30 Digital Elevation Model: US Geological Survey, 0.5-minute Global Topgography grid.

Wessel, P., and Smith, W. H. F., 1991, Free software helps map and display data: EOS Transactions, American Geophysical Union, v. 72, no. 41, p. 441, 445-446.

Appendix 1: Python script I

Procedure for generating the 'base grid' from all available source data

```
# -*- coding: cp1252 -*-
##########################
# GEBCO Cook Book Appendix 1  "Stage 1: generating a base grid for the International Bathymetric Chart of the Arctic Ocean (IBCAO) Version 3.0"
#
# This script contains the procedure used in stage 1 of the IBCAO 3.0 grid generation process. It covers the generation
# of a base grid based on all the available bathymetric source data. Stage 2 of the process involves adding high density data
#on top of the base grid using the 'remove-restore' procedure. This is covered in a separate script.
#
#This script is setup with example values included. The example values were used to grid IBCAO 3.0.
# The script will perform the following:
# 1. Setup of variables
# 2. Creation of file list for input to gridding
# 3. Check data consistency and merge files
# 4. Segment the data set into regions to handle large data amounts with GMT "blockmedian"
# 5. Block median filtering
# 6. Merge block median filtered segments into one file
# 7. Gridding, crop, filter, and resampling
#       7.1. Gridding using surface.
#       7.2. Crop and filter the grid
#       7.3 Resampling of the grid.
#
# Written by Martin Jakobsson
#Department of Geological Sciences
# Stockholm University
#
# September 23, 2012
#
#
#
#
#######################################################################
#################

from __future__ import division #This line includes the true division avoiding problems with unwanted floor division
import fileinput
import os

###########################
# 1: Setup of variables
###########################
Cint=0 #Counter for loop in section 3
DIRstr='C:\\IBCAOData\\'  #Path to data files
DATAstr='IBCAOinput.xyz' #Main file holding the source data for gridding. It consists of a merge of all input files.

# GMT input variables
BM_I='2000' # Blockmedian filter bin size, this should be the same as will be used for the gridding work,
BM_R='-2910000/2910000/-2910000/2910000' #Blockmedian region set for GMT. This is the regional set in Polarstereographic coordinates for the IBCAO gridding.
#The user must change this to the appropriate region for the task. It could be set in any coordinate system or lat lon if a geographic grid is created. Note that this region should extend slightly outside of the final grid boundaries to account for edge effects.

BM_R1='0/2910000/0/2910000' #Region 1 for subset tiling. This tiling is only required for very large data sets (100 million soundings or more).
BM_R2='0/2910000/-2910000/0' #Region 2 for subset tiling
```

```
BM_R3='-2910000/0/-2910000/0' #Region 3 for subset tiling
BM_R4='-2910000/0/0/2910000' #Region 4 for subset tiling

#Output files from the subset tiling
R1='R1.xyz'
R2='R2.xyz'
R3='R3.xyz'
R4='R4.xyz'

#Output files from the Blockmedian filtering of the different regions
BM_OUTPUT1='OUTFILE_BM1.xyz'
BM_OUTPUT2='OUTFILE_BM2.xyz'
BM_OUTPUT3='OUTFILE_BM3.xyz'
BM_OUTPUT4='OUTFILE_BM4.xyz'
BM_OUTPUT='OUTFILE_BM.xyz' #Output file of the entire region

S_I='2000' #Gridding cell size. Value is set as used for IBCAO 3.0
R_I='500' #Resampling grid size
FILT='6000' #Filter size for smoothing
S_T='0.32' #Tension parameter, for use with GMT's surface
N_I='5000' #Max iterations in each gridding cycle. Default is set to 250
LU='5000' #Value constraining the upper limit of the grid
S_grd_out='out_grd.grd' #Grid output file, unfiltered
CUT_R='-2904000/2904000/-2904000/2904000' #Geographic region of the final grid
CUT_grd_out='out_grd01.grd' #Cropped grid
FILT_grd_out='filt_grd01.grd' #Filtered grid
RESAMP_grd_out='resamp_grd01.grd' #Resampled grid at the spacing set by R_I


##########################
# Point 2: Creation of filelist for input to the gridding procedure
# Note that there is not a separation of the data here into different categories, this is done in the remove-restore procedure
##########################
#Input files are listed below, for additional files increase the list
FILElist=['file1_SB.xyz', 'file2_MB.xyz', 'file3_CONT.xyz','file4_TOPO.xyz']
print FILElist[1]

###########################
# Point 7: Gridding , crop, filter and resampling
###########################
def grid():

   #############################################
   # 7.1. Grid the data using GMT surface
   os.system('surface %s\\%s -R%s -I%s -N%s -T%s -G%s\\%s -Lu%s -V' % (DIRstr, BM_OUTPUT, BM_R, S_I, N_I, S_T, DIRstr, S_grd_out, LU))

   #############################################
   # 7.2. Crop and filter the grid
   os.system('grdcut %s\\%s -R%s -G%s\\%s' % (DIRstr, S_grd_out, CUT_R, DIRstr, CUT_grd_out))
   os.system('grdfilter %s\\%s -G%s\\%s -R%s -D0 -Fc%s' % (DIRstr, CUT_grd_out, DIRstr, FILT_grd_out, CUT_R, FILT))


   ##############################################
   # 7.3 Resampling of the IBCAO grid
   os.system('grdsample %s\\%s -G%s\\%s -R%s -I%s -V' % (DIRstr, FILT_grd_out, DIRstr, RESAMP_grd_out, CUT_R, R_I))

   ##############################################
```

```
   exit()

###########################
# Point 3: Check the data for consistency (no empty lines or large outliers) and file merge
###########################

F=open(r'%s\\%s' % (DIRstr,DATAstr), 'w+' )


print 'input file directory %s\n' % (DIRstr)
print '1. The following files will scanned and checked for consistency\n'
print '%s' % (FILElist)


while Cint <= len(FILElist)-1:
   G=open('%s%s' % (DIRstr,FILElist[Cint]), 'r' )
   while 1:
      line=G.readline()
      line_items=line.split()
      if not line: break
      if len(line_items) == 3:
         F.write(line)
   Cint +=1
   G.close()

F.close()


###############################
# Point 4: Segment the data set into regions to handle large data amounts with GMT "blockmedian"
###############################


os.system('set EMM=RAM')
os.system('set dos=high, umb')
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R1,DIRstr, DATAstr, DIRstr, R1))
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R2,DIRstr, DATAstr, DIRstr, R2))
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R3,DIRstr, DATAstr, DIRstr, R3))
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R4,DIRstr, DATAstr, DIRstr, R4))

# ##############################
# Point 5: Block median filtering
# ##############################

os.system('set EMM=RAM')
os.system('set dos=high, umb')
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R1,BM_I, DIRstr, R1, DIRstr,
BM_OUTPUT1))
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R2,BM_I, DIRstr, R2, DIRstr,
BM_OUTPUT2))
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R3,BM_I, DIRstr, R3, DIRstr,
BM_OUTPUT3))
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R4,BM_I, DIRstr, R4, DIRstr,
BM_OUTPUT4))
#os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R,BM_I, DIRstr, DATAstr, DIRstr, BM_OUTPUT))
#If only one region is used, the lines above should be commented out and this line used instead.

############################################
# Point 6 in the explanation. Merge all the block median data into one file
############################################
I=open(r'%s\\%s' % (DIRstr,BM_OUTPUT), 'w+' )
H=open('%s%s' % (DIRstr,BM_OUTPUT1), 'r' )
while 1:
```

```
    line=H.readline()
    if not line: break
    I.write(line)
H.close()

J=open('%s%s' % (DIRstr,BM_OUTPUT2), 'r' )
while 1:
    line=J.readline()
    if not line: break
    I.write(line)
J.close()

K=open('%s%s' % (DIRstr,BM_OUTPUT3), 'r' )
while 1:
    line=K.readline()
    if not line: break
    I.write(line)
K.close()

L=open('%s%s' % (DIRstr,BM_OUTPUT4), 'r' )
while 1:
    line=L.readline()
    if not line: break
    I.write(line)
L.close()

I.close()


#########################
# Point 7 in the explanation, see function in the beginning of the script.
#########################

grid()
```

Appendix 2: Python script II: the remove-restore procedure

The following script details the procedure for adding the higher resolution source data sets, such as multibeam, dense single beam echo soundings and land topography on top of the 'base grid' using the remove-restore method

The script is written in Python and uses embedded GMT routines.

```
# -*- coding: cp1252 -*-
#
# GEBCO Cook Book Appendix 2 "Step 2 in the process of gridding the International Bathymetric Chart of the Arctic Ocean (IBCAO) Version 3.0"
#
# This script is stage two in the procedure used in generating IBCAO 3.0. It uses embedded GMT routines.
# The script uses a number of variables which are setup in Point 8. These example values were used to grid
#IBCAO 3.0. The user should modify these variables to match their own gridding project.
#
# The script will perform the following:
# 8. Setup of the variables
# 9. Creation of an input file for gridding
# 10. Segment the data into regions to handle large data amounts with GMT "blockmedian"
# 11. Block median filtering
# 12. Merge block median filtered segments into one file
# 13. Find the depth in the 'base grid' at the location of the block median filtered data points
# 14. Compute the difference in z-values between the block median data and the base grid
# 15. Create a difference grid
# 16. Merge the difference grid on top of the base grid
#
#This script originated from the remove-restore script written by D. Sandwell 08/08/07
# Written by Martin Jakobsson November 27 2011
#
#
#
#
from __future__ import division #This line includes the true division avoiding problems with unwanted floor division
import fileinput
import os

##########################
# Point 8: Setup of the variables
#############################

DIRstr='C:\\IBCAOData\\'  #Directory path to the data files, change to your path
BM_R='-2904000/2904000/-2904000/2904000' #Blockmedian region set in GMT, this must fit the geographic extent of the grid which is going to be updated
```

```
#It could be any coordinate system, the main point is that the region fits the geographic region of the base grid to be updated

#The following are the output files from the Blockmedian filtering from each of the segmented regions for use in remove restore
BM_RR_OUTPUT1='OUTFILE_RR_BM_1.xyz'
BM_RR_OUTPUT2='OUTFILE_RR_BM_2.xyz'
BM_RR_OUTPUT3='OUTFILE_RR_BM_3.xyz'
BM_RR_OUTPUT4='OUTFILE_RR_BM_4.xyz'

BM_I='500' # Blockmedian filter bin size, this should be the same as the cell size of the resampled base grid.
IBCAO_resamp='resamp_grd01.grd' #The base grid
IBCAO_RR='IBCAO_RR.xyz'  #The input ASCII file of all multibeam and dense single beam echo soundings and land topography.
IBCAO_RR_SEGMENT='IBCAO_RR_SEGMENT.xyz' #This is a file that consists of all merged blockmedian files.

IBCAO_resamp_predict='resamp_grd01_pred.xyzz' #This is the interpolated values output from grdtrack
IBCAO_resamp_diff='IBCAO_diff.xyz' #Points that are different when from the old IBCAO after check with grdtrack against the new data
IBCAO_resamp_diff_GRD='IBCAO_diff.grd'
IBCAO_final='IBCAO_final_RR.grd' #The final grid file
DIFF_P='0.5' #the difference criteria that need to be passed for updating of grid. In other words, the data sets need to be different by at least this amount for an update to take place

S_C='0.001' #Iteration parameter
S_T='0.32' #Tension variable for use with the GMT surface gridding command

#In order to handle large volumes of data with Blockmedian, it may be necessary to cut the full area into sections. The user must change the values given here to those appropriate for their own gridding project. Values can be set in any coordinate system or latitude or longitude if a geographic grid is created. Note that this region should extend slightly outside of the final grid boundaries to account for edge effects.
BM_R1='0/2910000/0/2910000' #Region 1 for subset tiling. This tiling is only required for very large data sets (100 million soundings or more).
BM_R2='0/2910000/-2910000/0' #Region 2 for subset tiling
BM_R3='-2910000/0/-2910000/0' #Region 3 for subset tiling
BM_R4='-2910000/0/0/2910000' #Region 4 for subset tiling

#Output files from the subset tiling
R1='R1_RR.xyz'
R2='R2_RR.xyz'
R3='R3_RR.xyz'
R4='R4_RR.xyz'

######################################
# Point 9: Creation of an input file for gridding
######################################
```

```
os.system('set EMM=RAM')
os.system('set dos=high, umb')
DATAstr=IBCAO_RR
FILElist=["file1_SB_dense.xyz','file2_MB.xyz','file4_TOPO.xyz"] #All input files should be listed here.
print '%s' % (FILElist)
F=open(r'%s\\%s' % (DIRstr,DATAstr), 'w+' )
Cint=0
while Cint <= len(FILElist)-1:
   G=open('%s%s' % (DIRstr,FILElist[Cint]), 'r' )
   while 1:
      line=G.readline()
      line_items=line.split()
      if not line: break
      if len(line_items) == 3:
         F.write(line)
   Cint +=1
   G.close()

###############################################################################
# Point 10: Segment the data into regions to handle large data amounts with GMT "blockmedian"
###############################################################################

os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R1,DIRstr, DATAstr, DIRstr, R1))
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R2,DIRstr, DATAstr, DIRstr, R2))
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R3,DIRstr, DATAstr, DIRstr, R3))
os.system('gmtselect -R%s -V %s\\%s > %s\\%s' % (BM_R4,DIRstr, DATAstr, DIRstr, R4))

############################
# Point 11: Block median filtering
############################

os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R1,BM_I, DIRstr, R1, DIRstr,
BM_RR_OUTPUT1))
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R2,BM_I, DIRstr, R2, DIRstr,
BM_RR_OUTPUT2))
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R3,BM_I, DIRstr, R3, DIRstr,
BM_RR_OUTPUT3))
os.system('blockmedian -R%s -I%s -V -Q %s\\%s > %s\\%s' % (BM_R4,BM_I, DIRstr, R4, DIRstr,
BM_RR_OUTPUT4))
F.close()

#############################################
# Point 12: Merge the blockmedian filtered segments
#############################################

DATAstr2=IBCAO_RR_SEGMENT
L=open(r'%s\\%s' % (DIRstr,DATAstr2), 'w+' )
FILElist2=['OUTFILE_RR_BM_1.xyz','OUTFILE_RR_BM_2.xyz','OUTFILE_RR_BM_3.xyz','OUTFILE_RR_B
M_4.xyz']
Cint2=0
while Cint2 <= len(FILElist2)-1:
   K=open('%s%s' % (DIRstr,FILElist2[Cint2]), 'r' )
```

```
    while 1:
        line=K.readline()
        line_items=line.split()
        if not line: break
        if len(line_items) == 3:
            L.write(line)
    Cint2 +=1
    K.close()

L.close()

##################
# Point 13: Find the depth in the 'base grid' at the location of the blockmedian filtered data points
##################
#Find the depth in the base grid at the location of the blockmedian filtered data points.

os.system('grdtrack %s -G%s -V -S > %s' %
(IBCAO_RR_SEGMENT,IBCAO_resamp,IBCAO_resamp_predict))

################################################################################
# Point 14: Compute the difference in z-values between the blockmedian data and the base grid
#################################################################################

I=open('%s' % (IBCAO_resamp_diff), 'w+' )
H=open('%s' % (IBCAO_resamp_predict), 'r' )
while 1:
    line2=H.readline()
    if not line2: break
    X_str,Y_str,Z1_str,Z2_str=line2.split()
    X_f=float(X_str)
    Y_f=float(Y_str)
    Z1_f=float(Z1_str)
    Z2_f=float(Z2_str)
    DIFF_1=Z2_f-Z1_f
    DIFF_2=Z1_f-Z2_f
    DIFF_P_f=float(DIFF_P)
    if DIFF_1 > DIFF_P_f or DIFF_2 > DIFF_P_f:
        I.write('%s %s %s\n' % (X_str,Y_str,DIFF_2))
I.close()

###############################
# Point 15: Create a difference grid
###############################

#Create a grid using the GMT nearneighbor routine, that consist of the differences between the blockmedian
filtered data and the base grid

os.system('nearneighbor %s -G%s -R%s -I%s -S1000 -N4 -E0 -V' %
(IBCAO_resamp_diff,IBCAO_resamp_diff_GRD,BM_R,BM_I))

#################################################################
# Point 16: Merge the difference grid on top of the base grid
```

```
##################################################################

#This step adds the difference grid on top of the base grid using grdmath from GMT

os.system('grdmath %s %s ADD = %s' % (IBCAO_resamp,IBCAO_resamp_diff_GRD,IBCAO_final))

###
###  clean up the mess
###
##rm global.grd diff.grd



import fileinput
```

Appendix 3: IBCAO Compilation Team

[1]Martin Jakobsson, [2]Larry Mayer, [3]Bernard Coakley, [4]Julian A. Dowdeswell, [5]Steve Forbes, [6]Boris Fridman, [7]Hanne Hodnesdal, [8]Riko Noormets, [9]Richard Pedersen,[10]Michele Rebesco [11]Hans-Werner Schenke, [12]Yulia Zarayskaya, [10]Daniela Accettella, [2]Andrew Armstrong, [13]Robert M. Anderson, [14]Paul Bienhoff, [15]Angelo Camerlenghi, [16]Ian Church, [17]Margo Edwards, [2]James V. Gardner; [18]John K. Hall, [1]Benjamin Hell, [19]Ole Hestvik, [20]Yngve Kristoffersen, [21]Christian Marcussen, [1]Rezwan Mohammad, [2]David Monahan, [22]David Mosher, [23]Son V. Nghiem, [15]Maria Teresa Pedrosa, [5]Paola G. Travaglini, [24]Pauline Weatherall;

[1]Dept. of Geological Sciences, Stockholm University, Sweden; [2]Center for Coastal and Ocean Mapping/Joint Hydrographic Center, University of New Hampshire, USA; [3]Dept. of Geology and Geophysics, University of Alaska Fairbanks, USA; [4]Scott Polar Research Institute, University of Cambridge, UK; [5]Canadian Hydrographic Service, Canada; [6]Moscow Aerogeodetic Company, Russian Federation; [7]Norwegian Mapping Authority, Hydrographic Service, Norway; [8]The University Centre in Svalbard, Longyearbyen, Norway; [9]National Survey and Cadastre, Denmark; [10]Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS), Italy; [11]Alfred Wegener Institute for Polar and Marine Research (AWI), Germany; [12] Laboratory of Ocean Floor Geomorphology and Tectonics, Geological Institute RAS, Russian Federation; [13]Science Applications International Corporation, USA; [14] Johns Hopkins University Applied Physics Laboratory, USA;[15]ICREA and University of Barcelona, Spain; [16]Dept. Geodesy and Geomatics Engineering, University of New Brunswick, Canada;[17]University of Hawaii at Manoa, USA;[18]Geological Survey of Israel, Israel;[19]OLEX, Norway;[20]Dept of Earth Science, University of Bergen, Norway;[21]Geological Survey of Denmark and Greenland, Denmark;[22]Geological Survey of Canada, Canada[23]Jet Propulsion Laboratory, California Institute of Technology, USA; [24]British Oceanographic Data Centre (BODC), UK.

8.3 Testing of gridding methods

In order to test different gridding methods, it is necessary to compare the resulting grid with control data that were not used to make the grid. These control data may have been intentionally withheld for testing purposes, or they may not have been available when the grid was calculated. The differences between the control data and the grid are considered to be "errors," which can be quantified and compared to assess how well a gridding method does.

8.3.1 Distance to Control

Contributed by K. M. Marks, NOAA Laboratory for Satellite Altimetry, USA

We have developed a method that can be used to test different gridding algorithms. It has been used previously to assess errors in bathymetry models (Marks and Smith, 2010). The method can also be used to assess errors in any local or regional depth grids where there are multibeam or other "ground truth" depths available to compare the grids against. The "ground truth" data must not be incorporated into the depth grid for the technique to work, and there needs to be an area with a large gap between control points for testing. This method can also be used to evaluate different gridding techniques. A successful gridding algorithm will have smaller errors when compared to the ground truth data.

We demonstrate the use of the error assessment method to evaluate different gridding algorithms. For this example, following NOAA Technical Report NESDIS 132 (Marks and Smith, 2011), we select a small study area encompassing several seamounts (left panel in Figure 8.7) that has large gaps between control points and is traversed by several swaths of JAMSTEC multibeam data that can serve as "ground truth." Figure 8.7 (right panel) shows a distance to control map that was calculated from control points in bathymetry model version 12.1*. Version 12.1* had JAMSTEC data intentionally withheld from bathymetry model version 12.1 (Smith and Sandwell, 1997) for testing purposes.

Distance to control may be calculated in two ways. Smith wrote a program (Appendix C; Marks and Smith, 2011) exploiting the even/odd encoding of control points in Smith and Sandwell bathymetry models to compute a distance to control grid directly from the model "img" file. However, distance to control may be calculated from any control points using the GMT routine "grdmath" with the PDIST operator. The command line for calculating distance to control from any user-supplied control points is listed below. This will enable users to test any bathymetry grids for errors as a function of distance to control.

```
grdmath controlpoints.xy –Rminlon/maxlon/minlat/maxlat –Igrdspacing –fg PDIST = distance_grd
```

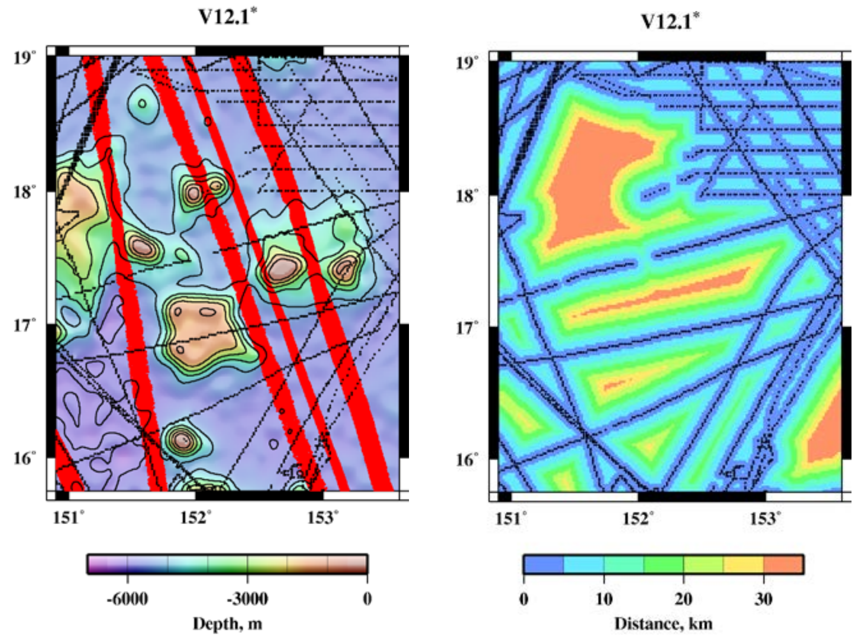


Figure 8.7 JAMSTEC multibeam swaths (red) plotted on bathymetry version 12.1* (left). Map of "distance to control" corresponding to bathymetry version 12.1* (right). Black dots are V12.1* grid cells constrained by ship soundings. V12.1* was created without JAMSTEC data.

In our example here we used depth values from the constrained grid cells from V12.1* in the study area as input into different gridding algorithms: GMT routine "surface," employed with the tension set to "0" and set to "1," and GMT routine "nearneighbor." Surface tension set to "0" gives the minimum curvature solution, and set to "1" gives a harmonic surface where maxima and minima are only possible at control points. "Nearneighbor" uses a nearest neighbor algorithm to assign an average value within a radius centered on a node. The GMT command lines used to produce the gridding solutions shown in Figure 8.8 are listed below.

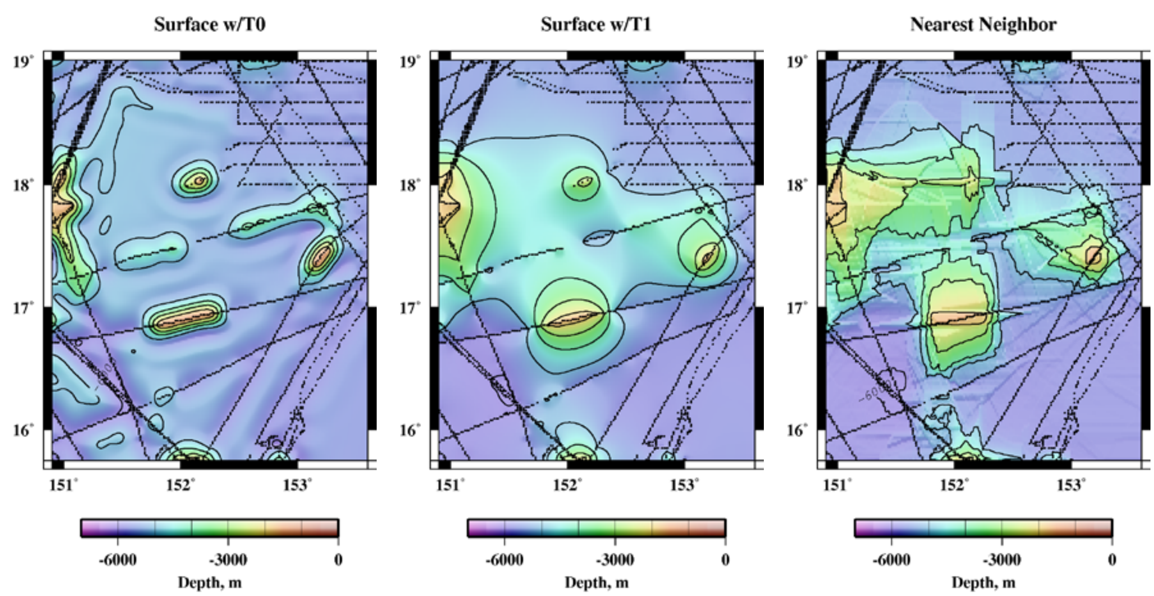


Figure 8.8 Results of gridding depths from V12.1* at constrained grid cells (black dots): GMT "surface" gridding routine with tension set to 0 (left) and set to 1 (middle), and "nearneighbor" gridding routine (right).

```
surface topo_12.1.nojamstec.controls.xyz -R0/2.7/0/3.41666666667 -I1m -T0 -Gsurface.t0_grd
surface topo_12.1.nojamstec.controls.xyz -R0/2.7/0/3.41666666667 -I1m -T1 -Gsurface.t1_grd
nearneighbor topo_12.1.nojamstec.controls.xyz -R0/2.7/0/3.41666666667 -N4/1 -S100k -I1m
        -Gnearneighor.1m_grd
```

Each gridding algorithm produces a very different result. By comparing these results to the JAMSTEC multibeam "ground truth" data (red dots in Fig. 8.7, left panel), we can assess which algorithm produced the best match to the observed depths. Note that no interpolated solution detects seamounts in gaps between ship controls (Fig. 8.8). However altimetric bathymetry model V12.1* (Fig. 8.7, left panel) does detect seamounts in gaps between ship soundings. The model fills these gaps with depths estimated from satellite gravity, which reflects the underlying seafloor topography.

In Figure 8.9 we show the errors (i.e., the absolute value of the differences) between the JAMSTEC multibeam depths and depths produced by the different GMT gridding routines. In our example, GMT routine "surface" with a tension of 1 has the smallest errors.

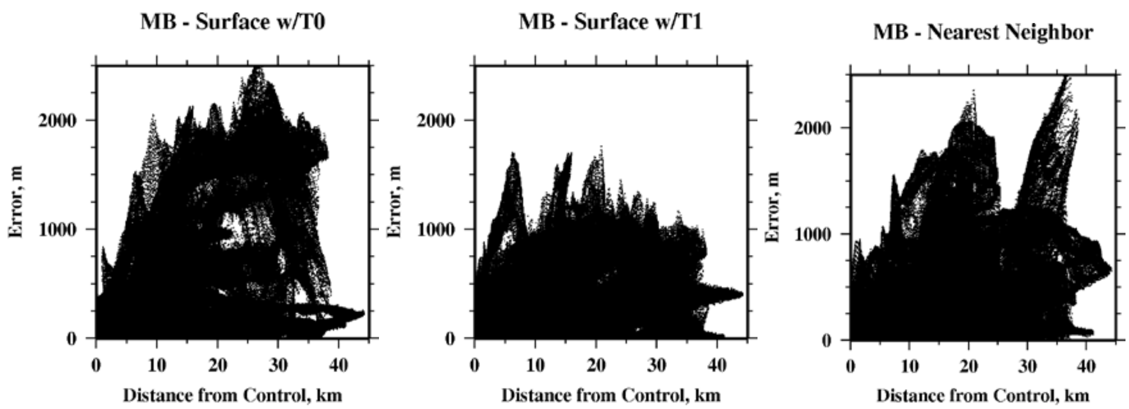


Figure 8.9 Errors are the absolute value of the differences between multibeam depths from swaths plotted in Figure 8.7 and gridded depths from GMT routines "surface" with tension set to 0 (left) and set to 1 (middle), and from GMT "nearneighbor" (right), plotted against distance from the nearest sounding control.

8.3.2 Histograms

The errors can also be displayed in the form of a histogram (Figure 8.10). The error distribution from gridding with GMT routine "surface" with tension set to 0 shows the largest errors and they are generally negative (depths from "surface" grid are deeper than multibeam depths).

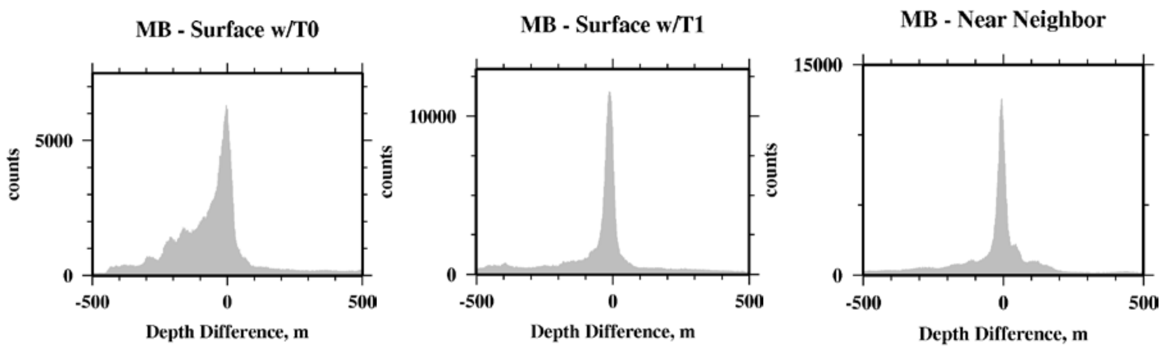


Figure 8.10 Histogram of the differences between multibeam depths from swaths plotted in Figure 8.7 and grids from GMT routines "surface" with tension set to 0 (left) and set to 1 (middle), and from GMT "nearneighbor" (right) depths.

It is also possible to map regional errors resulting from the different gridding algorithms. The interpolated grids (shown in Fig. 8.8) can be subtracted from a gridded solution that incorporated all the ship controls (in this example, bathymetry model V12.1). The pattern of regional depth differences (Fig. 8.11) can help in evaluating how well a gridding routine is doing.

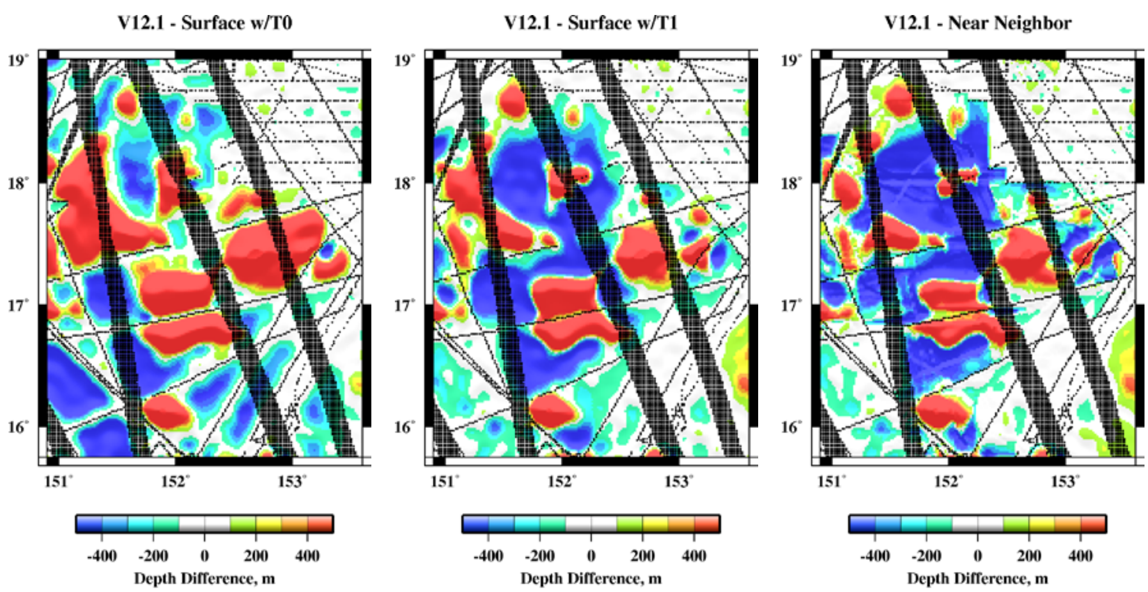


Figure 8.11 Regional depth differences between V12.1 and results from different GMT gridding routines (shown in Fig. 8.8). Black dots are constrained grid cells in V12.1.

REFERENCES

Marks K, Smith WHF (2010) Evolution of errors in the altimetric bathymetry model used by Google Earth and GEBCO. Mar. Geophys. Res, 31:223-238, doi:10.1007/s11001-010-9102-0

Marks K, Smith WHF (2011) Assessing errors in altimetric and other bathymetry grids. NOAA Technical Report NESDIS 132, January, 2011

Smith WHF, Sandwell DT (1997) Global sea floor topography from satellite altimetry and ship depth soundings. Science 277:1956-1962. doi:10.1126/science.277.5334.1956

8.3.3 Statistical tests

TEXT IN PREPARATION

8.4 Other routines for use on gridded data

Contributed by Matthew Love and Barry Eakins, National Geophysical Data Center, USA, and David Sandwell, Scripps Institution of Oceanography, USA

Below are miscellaneous routines for use on gridded data. The routines include gdal_query.py, which compares gridded elevations with xyz data, update_grid, which updates a grid with new grid cell values, and gdal_perspective.py, which creates color-perspective images of DEMs.

gdal_query.py

This Python script uses GDAL (http://www.gdal.org/) to query a GDAL-readable grid, such as netcdf, at the locations of xyz data in a specified file. It returns an ASCII file with xyz of the original data, as well as the grid elevation value at each point and the difference between these values. This script is useful for quantifying how well the grid represents the source data it was built from, or compares with other, independent data that were not used in grid development. Developer - Matthew Love, NOAA National Geophysical Data Center.

```
#!/usr/bin/env python

 # -
 # Description: Compare the values found in a source 'xyz' ascii
 #              file to those of a gdal compatible gridfile, or
 #              sample a gridfile to xy points in an xy* file.
 #
 # Depends: GDAL ; NumPy
 # -

#--
import sys
import os
import math
import re
import struct
import numpy as np
import osgeo.gdal as gdal
#--

gq_version = '1.8.2'

def con_dec(x, dec):
    '''Return a float string with n decimals
    (used for ascii output).'''

    if x is None:
        print >> sys.stderr, "gdal_query: Error, Attempting to convert a 
'None' value."
        return
    fstr = "%." + str(dec) + "f"
    return fstr % x

# Convert a geographic x,y value to a pixel location of geoTransform
def geo2pixel( geo_x, geo_y, geoTransform ):
    if geoTransform[2] + geoTransform[4] == 0:
        pixel_x = (geo_x - geoTransform[0]) / geoTransform[1]
        pixel_y = (geo_y - geoTransform[3]) / geoTransform[5]
    else:
```

```
        pixel_x, pixel_y = apply_gt( geo_x, geo_y, invert_gt( geoTransform
) )
    return int(pixel_x), int(pixel_y)

# Convert a pixel location to geographic coordinates given geoTransform
def pixel2geo( pixel_x, pixel_y, geoTransform ):
    geo_x, geo_y = apply_gt( pixel_x, pixel_y, geoTransform )
    return geo_x, geo_y

def apply_gt( in_x, in_y, geoTransform ):
    out_x = geoTransform[0] + in_x * geoTransform[1] + in_y *
geoTransform[2]
    out_y = geoTransform[3] + in_x * geoTransform[4] + in_y *
geoTransform[5]
    return out_x, out_y

def invert_gt(geoTransform):
    det = geoTransform[1] * geoTransform[5] - geoTransform[2] *
geoTransform[4]
    if abs(det) < 0.000000000000001:
        return
    invDet = 1.0 / det
    # compute adjoint and divide by determinate
    outGeoTransform = [0,0,0,0,0,0]
    outGeoTransform[1] = geoTransform[5] * invDet
    outGeoTransform[4] = -geoTransform[4] * invDet
    outGeoTransform[2] = -geoTransform[2] * invDet
    outGeoTransfrom[5] = geoTransform[1] * invDet
    outGeoTransform[0] = ( geoTransform[2] * geoTransform[3] -
geoTransform[0] * geoTransform[5] ) * invDet
    outGeoTransform[3] = ( -geoTransform[1] * geoTransform[3] +
geoTransform[0] * geoTransform[4] ) * invDet
    return outGeoTransform

def
query_gdal(inxyz,ingrd,delim,xloc,yloc,zloc,nodata,out_form,addit,return_a
ll,verbose):
    '''Compare an xyz file to a gdal-compatible grid file.'''

    if inxyz == sys.stdin:
        in_plot = inxyz
    else:
        in_plot = open(inxyz, 'r')

    # Process the grid file
    ds = gdal.Open(ingrd)
    comp_geot = ds.GetGeoTransform()
    band = ds.GetRasterBand(1)

    # Load the grid into a numpy array
    tgrid = band.ReadAsArray()
    #except:
xyz_vs_gdal(inxyz,ingrd,delim,xloc,yloc,zloc,nodata,out_form,addit,return_
all,verbose)

    # Get some grid info
    if nodata is None: nodata = band.GetNoDataValue()
    if nodata is None: nodata = -9999

    cellsize = [float(comp_geot[1]), float(comp_geot[5])]
    xextent = float(comp_geot[0])
```

```
    yextent = float(comp_geot[3])

    # Process the xyz file
    n = 0
    for i in in_plot:
        n+=1
        if verbose:
            print >> sys.stderr, "gdal_query: " + str(n), "\r",
        try:
            x = float(i.split(delim)[int(xloc)].strip())
            y = float(i.split(delim)[int(yloc)].strip())
            if zloc == '-':
                z = nodata
            else:
                z = float(i.split(delim)[int(zloc)].strip())

        except:
            print >> sys.stderr, "gdal_query: Failed to read line: %s"
%(i),
            x=xextent - 10
            y=yextent + 10

        # Continue if values are reasonable.
        if x > xextent and y < yextent:
            xpos,ypos = geo2pixel(x,y,comp_geot)
            #xpos = int(math.fabs(math.ceil((xextent -
x)/float(cellsize[0]))))
            #ypos = int(math.fabs(math.ceil((y -
yextent)/float(cellsize[1]*-1))))

            # Locate the grid cell and get it's value
            try: g = tgrid[ypos,xpos]
            except: g = nodata

            d = nodata
            c = nodata
            p = nodata
            s = nodata

            if g != nodata:
                d = z - g
                p = z + g
                # FIXME ## as d/g*100 woudl fail if g was zero
                if g == 0:
                    g += 0.0000001
                # /FIXME ##
                # c is the percent difference
                c = con_dec(math.fabs(float(d/g*100)), 2)
                # s is the 'scaled' difference
                s = con_dec(math.fabs(d / (z + g)), 4)

            # Print out the results
            d = con_dec(d, 4)

            outs = []
            for i in out_form:
                outs.append(vars()[i])
            print(delim.join(map(str, outs)))
        else:
            if return_all:
                d = nodata
```

```
                p = nodata
                c = nodata
                s = nodata
                g = nodata
                outs = []
                for i in out_form:
                    outs.append(vars()[i])
                print(delim.join(map(str, outs)))
    if verbose:
        print >> sys.stderr, "\n",

def Usage():
    print('Usage: gdal_query.py [-delimiter char] [-s_format "0,1,2"] ')
    print('                     [-d_format "xyzgdps"] [-d_nodata value] [-
header]')
    print('                     [-return_all] [-verbose]')
    print('                     grdfile [srcfile]')
    print('')
    print('Input options:')
    print('  grdfile\tSpecifies which gdal-compatible grid file in which
to ')
    print('  \t\tquery values.')
    print('  srcfile\tSpecifies which xy* file in which to query grdfile
with,')
    print('  \t\tif not given, will read from standard input.')
    print('  -delimiter\tSpecifies the input xy* delimiter')
    print('  -s_format\tSpecifies the input xy* format, a quoted triplet
specifying')
    print('  \t\twhere the x,y and z values are located.')
    print('  \t\tIf there is no input z value, specify with a -, i.e. ')
    print('  \t\t-s_format "0,1,-"')
    print('')
    print('Output options:')
    print('  -d_format\tSpecifies the output xy* format, a quoted string
specifying ')
    print('  \t\twhich output values to return.')
    print('  \t\tOptions are x,y,z,g,d,p,c,s; where z is the input z
value, g is the grid ')
    print('  \t\tz value, d is the difference between the input z value
and grid')
    print('  \t\tz value, p is the product of the input z value and the
grid z vallue,')
    print('  \t\tc is the percetage difference and s is the scaled
difference.')
    print('  -d_nodata\tSpecifies the output nodata value, will use the
nodata value from')
    print('  \t\tthe input grid if not specified.')
    print('  -header\tIndicates that the output will have a header
included, default is False.')
    print('  -return_all\tWill prompt the return of all input points,
regardless of location')
    print('  \t\twhen compared to the input grid file, with grid values
considered as nodata.')
    print('')
    print('General options:')
    print('  -verbose\tWill increase verbosity.')
    print('')
    print('Example:')
    print('To query grd.tif with an ascii xy file (values.xy) with comma
delimited fields and ')
```

```
    print('no z-value and return the xy values from the input xy file and
the z values')
    print('from grd.tif:')
    print('gdal_query.py -delimiter "," -s_format "0,1,-" -d_format "xyg"
grd.tif values.xy > values.xyz')
    print('\ngdal_query v.%s' %(gq_version))
    sys.exit( 1 )
#
# Mainline
#
if __name__ == "__main__":

    inxyz=None
    ingrd=None
    delim=" "
    xyzf="0,1,2"
    out_xyzf="xyzg"
    xyzh=False
    addit=False
    verbose=False
    return_all=False
    out_nodata=None

    gdal.AllRegister()
    argv = gdal.GeneralCmdLineProcessor( sys.argv )
    if argv is None:
        sys.exit(0)

    # Parse command line arguments.
    i = 1
    while i < len(argv):
        arg = argv[i]

        if arg == '-delimiter':
            delim = argv[i+1]
            i = i + 1

        elif arg == '-s_format':
            xyzf = str(argv[i+1])
            i = i + 1

        elif arg == '-d_format':
            out_xyzf = str(argv[i+1])
            i = i + 1

        elif arg == '-d_nodata':
            out_nodata = float(argv[i+1])
            i = i + 1

        elif arg == '-header':
            xyzh = True

        elif arg == '-addition':
            addit = True

        elif arg == '-verbose':
            verbose = True

        elif arg == '-return_all':
            return_all = True
```

```
        elif arg[0] == '-':
            Usage()

        elif ingrd is None:
            ingrd = arg

        elif inxyz is None:
            inxyz = arg

        else:
            Usage()

        i = i + 1

    if inxyz == None:
        inxyz = sys.stdin
    if ingrd == None:
        Usage()
        sys.exit(0)

    #-- Parse point locations.
    xloc = xyzf.split(",")[0].strip()
    yloc = xyzf.split(",")[1].strip()
    zloc = xyzf.split(",")[2].strip()

    out_form = tuple(re.findall(r'(\D)',out_xyzf))

    if xyzh == True:
        outs = []
        for i in out_form:
            outs.append(i)
        print delim.join(outs)


query_gdal(inxyz,ingrd,delim,xloc,yloc,zloc,out_nodata,out_form,addit,retu
rn_all,verbose)
#--END
```

update_grid

This csh script uses GMT (http://gmt.soest.hawaii.edu/) to update a GMT readable grid with new data from a specified xyz file. The new grid created will have elevation values representative of the new data, not the values in the original grid. Cells without new data will retain the values of the original grid. This script is useful for quickly improving a pre-compiled grid with new, trusted data. Developer – David Sandwell, Scripps Institution of Oceanography.

```
#!/bin/csh
#  D. Sandwell 08/08/07
#
#  Update the topography grid using some new trusted data.
#
# check the number of arguments
#
 if ($#argv < 9) then
  echo " Usage: update_grid topo.xyz topo_old.grd lon0 lonf lat0 latf dlon
dlat edit"
  echo "        topo.xyz   -  file of lon, lat, depth (neg meters)"
  echo "        topo_old.grd - existing low resolution grid of topography"
  echo "        lon0       -  left boundary of output grid (0-360)"
  echo "        lonf       -  right boundary of output grid (0-360)"
  echo "        lat0       -  lower boundary of output grid"
  echo "        latf       -  upper boundary of output grid"
  echo "        dlon       -  longitude spacing of output grid (e.g.,
.001)"
  echo "        dlat       -  latitude spacing of output grid"
  echo "        edit       _  remove |differences| > edit"
  echo " "
  echo "Example: update_grid german_iceland.xyz topo_old.grd 340 343 65 68
.01 .01"

  exit 1
 endif
#
#  first blockmedian the input data to make the file smaller
#
blockmedian $1 -R$3/$4/$5/$6 -I$7/$8 > block.xyz
#
#   make a matching grid from the global topo
#
grd2xyz $2 -R$3/$4/$5/$6 -S > global.xyz
surface global.xyz -R$3/$4/$5/$6 -I$7/$8 -T.00 -V -Gglobal.grd
#
#  interpolate the new data through the old grid
#
grdtrack block.xyz -Gglobal.grd -S > predict.xyzz
#
awk '{if(!(($4-$3) > '$9' || ($3-$4) > '$9')) print($1, $2, $3-$4)}' <
predict.xyzz > block.xyd
#
#   now make the difference grid
#
surface block.xyd -R$3/$4/$5/$6 -I$7/$8 -T.35 -V -Gdiff.grd
#
#   add the two grids
#
```

```
grdmath  global.grd diff.grd ADD = final_topo.grd
#
#  clean up the mess
#
rm global.grd diff.grd
```

gdal_perspective.py

This Python script uses GDAL (http://www.gdal.org/), ImageMagick (http://imagemagick.org) and PovRay (http://povray.org) to generate a color perspective image of a GDAL-readable grid, such as netCDF or GeoTiff. This script is useful for generating quality perspective images of grids for use in DEM development as a tool for QA/QC and for reporting and visualization of the final DEM product. Developer - Matthew Love, NOAA National
Geophysical Data Center.

```
#!/usr/bin/env python

import sys
import os
import math
from osgeo import gdal
import numpy as np

gp_version = '0.0.2'

gg_equat = 111321.543

## Colors
trs = (-11000,-10500,-10000,-9500,-9000,-8500,-8000,-7500,-7000,
        -6500,-6000,-5500,-5000,-4500,-4000,-3500,-3000,-2500,-2000,
        -1500,-1000,-500,-0.001,0,100,200,500,1000,1500,2000,2500,3000,
        3500,4000,4500,5000,5500,6000,6500,7000,7500,8000)

colors =
([10,0,121],[26,0,137],[38,0,152],[27,3,166],[16,6,180],[5,9,193],

[0,14,203],[0,22,210],[0,30,216],[0,39,223],[12,68,231],[26,102,240],

[19,117,244],[14,133,249],[21,158,252],[30,178,255],[43,186,255],[55,193,2
55],

[65,200,255],[79,210,255],[94,223,255],[138,227,255],[138,227,255],

[51,102,0],[51,204,102],[187,228,146],[255,220,185],[243,202,137],[230,184
,88],

[217,166,39],[168,154,31],[164,144,25],[162,134,19],[159,123,13],[156,113,
7],

[153,102,0],[162,89,89],[178,118,118],[183,147,147],[194,176,176],[204,204
,204],
          [229,229,229],[138,227,255],[51,102,0])

elevs = [-20,-19.09090909,-18.18181818,-17.27272727,-16.36363636,-
15.45454545,
         -14.54545455,-13.63636364,-12.72727273,-11.81818182,-10.90909091,
         -10,-9.090909091,-8.181818182,-7.272727273,-6.363636364,-
5.454545455,
         -4.545454545,-3.636363636,-2.727272727,-1.818181818,-
0.909090909,-0.001,
         0,48.06865898,96.13731797,240.3432949,480.6865898,721.0298848,

961.3731797,1201.716475,1442.05977,1682.403064,1922.746359,2163.089654,
```

```
2403.432949,2643.776244,2884.119539,3124.462834,3364.806129,3605.149424,
         3845.492719]

def scale_el(value, min, max, tr):
    if value > 0 and max > 0:
        return (max * tr) / 8000
    elif value < 0 and min < 0:
        return (min * tr) / -11000
    elif value == 0:
        return 0
    else: return None

def make_cpt(ingrd, grdmm):
    out_file = "%s.cpt" %(ingrd)
    out_cpt = open(out_file, 'w')

    ## NoData
    nd = returnNoData(ingrd)
    nd_string = str(nd) + " 0 0 0\n"
    out_cpt.write(nd_string)

    for i,j in enumerate(elevs):
        elevs[i] = scale_el(elevs[i], grdmm[4], grdmm[5], trs[i])
        if elevs[i] != None:
            outs = elevs[i],colors[i][0],colors[i][1],colors[i][2]
            #print outs
            out_cpt.write(" ".join(map(str, outs)))
            out_cpt.write("\n")

## Get min/max info about `ingrd'
def returnMinMax(ingrd):
    # Process the grid file
    ds = gdal.Open(ingrd)
    comp_geot = ds.GetGeoTransform()

    cellsize = [float(comp_geot[1]), float(comp_geot[5])]
    xmin = float(comp_geot[0])
    ymax = float(comp_geot[3])
    #nodata = ds.GetRasterBand(1).GetNoDataValue()
    #print xmin,ymax

    band = ds.GetRasterBand(1)
    nodata = band.GetNoDataValue()
    tgrid = band.ReadAsArray()

    #tgrid[tgrid==nodata]=float('nan')
    #outarray[np.isnan(outarray)]=ndata
    if np.isnan(nodata):
        tgrid=np.ma.MaskedArray(tgrid, mask=(np.isnan(tgrid)))
    else:
        tgrid=np.ma.MaskedArray(tgrid, mask=(tgrid==nodata))

    #print comp_geot
    xs = ds.RasterXSize
    ys = ds.RasterYSize
    zmin = tgrid.min()
    zmax = tgrid.max()

    xmax = xmin + (cellsize[0] * xs)
    ymin = ymax + (cellsize[1] * ys)
```

```
    return [xmin, xmax, ymin, ymax, zmin, zmax, xs, ys]

def returnNoData(ingrd):
    ds = gdal.Open(ingrd)
    return ds.GetRasterBand(1).GetNoDataValue()

def degree2radian(deg):
    return (deg/180.0) * math.pi

def radian2degree(rad):
    return (rad/math.pi) * 180.0

def latlonlen(lat):
    lonlen = math.cos(degree2radian(lat)) * gg_equat
    latlen = 1 * gg_equat
    return [lonlen, latlen]

def pre_perspective(ingrd, grdmm, cpt):
    gd_translate = "gdal_translate -ot UInt16 -of PNG -scale %s %s 0 65535
%s temp.png\n\
gdal_translate -srcwin 1 1 %s %s -of PNG temp.png dem_16bit.png\n\
gdaldem color-relief %s %s temp2.tif\n\
gdal_translate -srcwin 1 1 %s %s temp2.tif rgb.tif\n\
rm temp.* temp2.*\n\
convert -transparent \"rgb(0,0,0)\" rgb.tif rgb.png\n\
convert -size 10000 dem_16bit.png dem_16bit_10000.png\n\
convert -transparent \"rgb(0,0,0)\" -size 10000 rgb.png rgb_10000.png\n"
%(grdmm[4], grdmm[5], ingrd,

grdmm[6]-2, grdmm[7]-2, ingrd,

cpt, grdmm[6]-2, grdmm[7]-2)
    return gd_translate

def povray_template(ingrd, grdmm, options):
    lllen = latlonlen(grdmm[2])
    povr_out = "%s.pov" %(ingrd)
    povr_file = open(povr_out, 'w')
    povr_file.write("// DEM\n\
\n\
//global_settings { assumed_gamma 2.2 }\n\
      \n\
#include \"colors.inc\"\n\
     \n\
#declare Bi = 2;\n\
\n\
//\n\
// Custom parameters start here\n\
//\n\
#declare rgb_image = \"rgb.png\"\n\
#declare dem_image = \"dem_16bit.png\"\n\
\n\
#declare xdim = %s;  //number of pixels in X-direction\n\
#declare ydim = %s;  //number of pixels in y-direction\n\
#declare max_y = %s; //maximum latitude extent\n\
#declare min_y = %s; //minimum latitude extent\n\
#declare min_z = %s; //minimum elevation\n\
#declare max_z = %s; //maximum elevation\n\
\n\
// Obtained from http://www.csgnetwork.com/degreelenllavcalc.html  \n\
#declare deg_lat_len = %s; //length of a degree of latitude in meters  \n\
```

```
#declare deg_lon_len = %s; //length of a degree of longitude in meters\n\
\n\
// Position of camera             \n\
#declare cam_azimuth = %s;\n\
#declare cam_elevation = %s;\n\
#declare cam_distance = %s; \n\
#declare cam_view_angle = %s;\n\
                     \n\
// Position of the \"sun\"  \n\
#declare light_azimuth = cam_azimuth+90;\n\
#declare light_elevation = %s;\n\
#declare light_distance = %s;             \n\
\n\
#declare vertical_exaggeration = %s;\n\
//\n\
// Custom parameters end here\n\
//\n\
\n\
#declare lon_scale = deg_lon_len / deg_lat_len;\n\
#declare z_scale = (100 * (max_z - min_z)) / (deg_lat_len * (max_y -
min_y));\n\
\n\
#declare cam_x = cam_distance * cos(radians(cam_elevation)) *
sin(radians(cam_azimuth));\n\
#declare cam_y = cam_distance * sin(radians(cam_elevation));\n\
#declare cam_z = cam_distance * cos(radians(cam_elevation)) *
cos(radians(cam_azimuth));\n\
#declare light_x = light_distance * cos(radians(light_elevation)) *
sin(radians(light_azimuth));\n\
#declare light_y = light_distance * sin(radians(light_elevation));\n\
#declare light_z = light_distance * cos(radians(light_elevation)) *
cos(radians(light_azimuth));\n\
\n\
#declare Texture0 = // Color elevation image (24-bit RGB PNG)\n\
texture {\n\
  pigment{\n\
    image_map { \n\
      png rgb_image map_type 0 once interpolate Bi \n\
    } \n\
  } \n\
  finish { ambient 0.4 diffuse 0.8 } \n\
  rotate x*90  \n\
}\n\
\n\
\n\
height_field { // Unsigned 16-bit PNG DEM\n\
   png dem_image \n\
   smooth\n\
   clipped_by {box { <0, 0, 0>, <0.999, 1, 0.999> } }\n\
   texture { Texture0 }\n\
   translate <-0.5, 0, -0.5>\n\
   scale <100*lon_scale*xdim/ydim,\n\
          vertical_exaggeration*z_scale,  //Vertical exaggeration\n\
          100>\n\
} \n\
\n\
\n\
camera {\n\
   angle cam_view_angle\n\
   location <cam_x, cam_y, cam_z>\n\
   look_at <0, 0, 0> \n\
```

```
}\n\
\n\
light_source { <light_x, light_y, light_z> color White shadowless parallel
}\n\
        \n\
background { White } \n\
" %(grdmm[6],grdmm[7],grdmm[3],
    grdmm[2],grdmm[4],grdmm[5],
    lllen[0],lllen[1],options[0],
    options[1],options[2],options[3],
    options[4],options[5],options[6]))
    povr_file.close()
#    print('\ngdal_perspective v.%s' %(gp_version))
def Usage():
    print('')
    print('gdal_perspective.py srcfile [-width n] [-height n]')
    print('                   [-azimuth n] [-elevation n]')
    print('                   [-distance n] [-view-angle n]')
    print('                   [-light-elevation n] [-light-distance n]')
    print('                   [-vertical-exaggeration n] [-cpt n] [-q]')
    print('')
    sys.exit( 1 )

if __name__ == "__main__":

    ingrd=None
    pov_wid=1000
    pov_height=800
    pov_azimuth=130
    pov_elevation=27
    pov_distance=235
    pov_view_angle=35
    pov_light_elevation=30
    pov_light_distance=10000
    pov_vertical_exag=2
    quiet=False
    user_cpt=None

    gdal.AllRegister()
    argv = gdal.GeneralCmdLineProcessor( sys.argv )
    if argv is None:
        sys.exit(0)

    # Parse command line arguments.
    i = 1
    while i < len(argv):
        arg = argv[i]

        if arg == '-width':
            pov_wid = sys.argv[i+1]
            i = i + 1

        elif arg == '-height':
            pov_height = sys.argv[i+1]
            i = i + 1

        elif arg == '-azimuth':
            pov_azimuth = sys.argv[i+1]
            i = i + 1

        elif arg == '-view-angle':
```

```
            pov_view_angle = sys.argv[i+1]
            i = i + 1

        elif arg == '-elevation':
            pov_elevation = sys.argv[i+1]
            i = i + 1

        elif arg == '-distance':
            pov_distance = sys.argv[i+1]
            i = i + 1

        elif arg == '-light-elevation':
            pov_distance = sys.argv[i+1]
            i = i + 1

        elif arg == '-light-distance':
            pov_distance = sys.argv[i+1]
            i = i + 1

        elif arg == '-vertical-exaggeration':
            pov_vertical_exag = sys.argv[i+1]
            i = i + 1

        elif arg == '-cpt':
            user_cpt = sys.argv[i+1]
            i = i + 1

        elif arg == '-q':
            quiet = True

        elif arg[0] == '-':
            Usage()

        elif ingrd is None:
            ingrd = arg

        else:
            Usage()

        i = i + 1

    if ingrd == None:
        Usage()
        sys.exit(0)


options=[pov_azimuth,pov_elevation,pov_distance,pov_view_angle,pov_light_e
levation,pov_light_distance,pov_vertical_exag]
    if not quiet:
        print("gdal_perspective: Obtaining min/max values from grid")
    grdmm = returnMinMax(ingrd)
    if not quiet:
        print("gdal_perspective: Generating CPT file")
    if user_cpt is None:
        make_cpt(ingrd, grdmm)
        pp_sh = pre_perspective(ingrd,grdmm,"%s.cpt" %(ingrd))
    else:
        pp_sh = pre_perspective(ingrd,grdmm,user_cpt)
    povray_template(ingrd, grdmm, options)
    povr_sh = "povray %s.pov +W%s +H%s" %(ingrd,pov_wid,pov_height)
    if not quiet:
```

```
        print("gdal_perspective: processing grid file")
    os.system(pp_sh)
    #print pp_sh
    if not quiet:
        print("gdal_perspective: processing pov-ray file")
    os.system(povr_sh)
    #print povr_sh
    # Cleanup
    os.system("rm rgb.* dem_*")
#--END
```

Chapter 9.0 Data Analysis Techniques

9.1 Computing Coherence on 2-D Grids

This chapter provides step-by-step instructions that will show a user how calculate the coherence between two-dimensional gravity and bathymetry grids.

Contributed by K. M. Marks, NOAA Laboratory for Satellite Altimetry, USA

9.1.1 What is Coherence?

Coherence is a measure of the linear correlation between two inputs. Coherence near one indicates a near perfect correlation of the two inputs, while coherence near zero indicates the absence of any significant relationship. A coherence of 0.5 can be interpreted as a signal-to-noise ratio of 1:1 in one input if the other can be assumed to be noise-free (Bendat and Piersol, 1986; Eq. 6.39). In the example here, the inputs are grids- one is multibeam bathymetry and the other marine gravity. One may expect a correlation between intermediate wavelength (~15-160 km) gravity and bathymetry anomalies, in accordance with isostatic compensation theory.

9.1.2 Data Grids

We demonstrate the coherence method on grids that lie on the northern flank of the Pacific-Antarctic spreading ridge and cover the Heirtzler fracture zone (Figure 9.1).

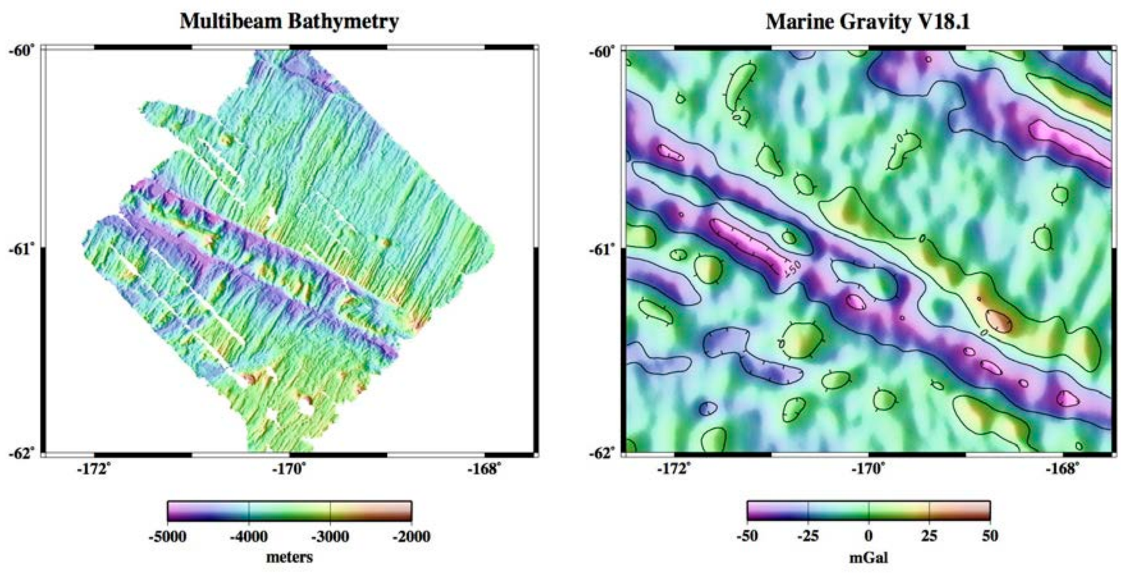


Figure 9.1 Multibeam bathymetry (left) and marine gravity (right) on the northern flank of the Pacific-Antarctic spreading ridge.

The multibeam bathymetry grid may be downloaded from the Lamont-Doherty Earth Observatory website using their Marine Geoscience Data System Geomapapp software (http://www.geomapapp.org/index.htm). The grid may be obtained on a geographic projection with a 200 m grid spacing.

The marine gravity anomaly grid (Sandwell and Smith, 1997, version 18.1) is available for download from the Scripps Institution of Oceanography website (http://topex.ucsd.edu/WWW_htlm/mar_grav.html). The gravity anomalies are derived from satellite altimeter data. The grid is on a Mercator projection with a 1-minute grid spacing.

9.1.3 Grid Preparation

The grids need to be prepared prior to computing the coherence. The grids are on different projections and have different grid spacings, and the multibeam survey is irregular in shape and inclined to the parallels and meridians. To correct for these irregularities and register the different data types, we sample the gravity grid at each multibeam point so that each data point record is longitude, latitude, depth, gravity (*xyzg*). We use GMT routine “grd2xyz” with the “-S” option to convert the multibeam grid to *xyz* points while suppressing points that have no depth value. We use “grdtrack” to interpolate the gravity grid to the multibeam point locations, thus obtaining *xyzg* for each data point record.

We correct for the inclined sides of the multibeam survey by projecting the *xyzg* points with an Oblique Mercator projection centered on the multibeam survey and rotating to maximize the rectangular extent of the coverage, via GMT routine “mapproject.” The projected and rotated points are shown in Figure 9.2. The projection routine reads positions in longitude and latitude, and outputs positions in meters from the projection center.

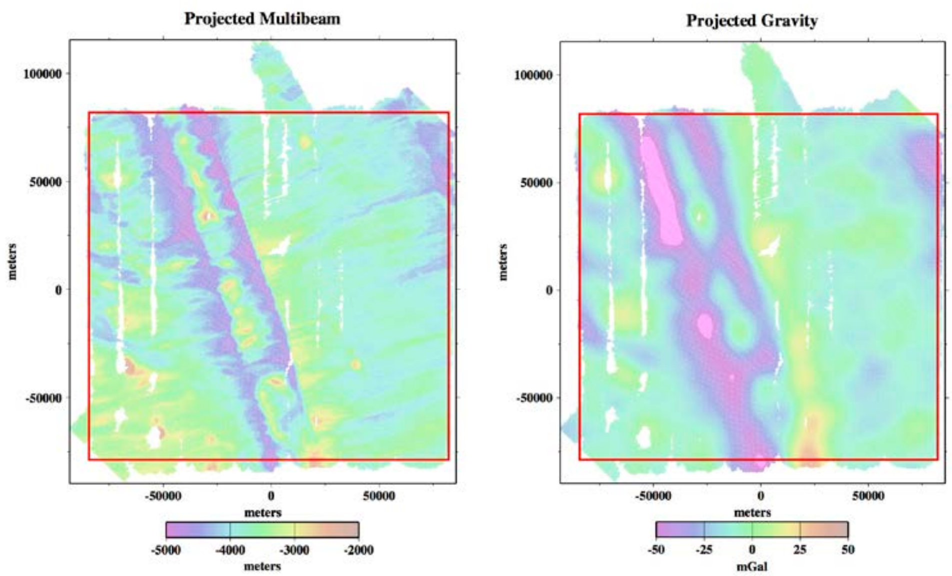


Figure 9.2 Oblique Mercator projected and rotated multibeam (left) and gravity (right) data points. Red box outlines trimming boundary.

The command lines used to project the data points shown in Figure 9.1 into the Oblique Mercator projected locations shown in Figure 9.2 follow. Note that routine "mapproject" is used three times- first, to bring the multibeam points (mb.xyz) into a Mercator projection matching that of the gravity grid (grav.18.1_grd) in order to sample it with "grdtrack;" second, to convert to geographic coordinates; and third, to project the data points into the Oblique Mercator projection including rotation (via the "–Joa" option).

```
gmtset ELLIPSOID Sphere
grd2xyz –S mb_grd > mb.xyz
mapproject mb.xyz -Jm1 -R-172.5/-167.5/-62.0001406129/-59.9969352129 | grdtrack
        -Ggrav.18.1_grd | mapproject -I -Jm1 -R-172.5/-167.5/-62.0001406129/-59.9969352129 |
        mapproject -C -F -Joa-170/-61/-134/1 -R-172.5/-167.5/-62/-60 > mb.grav.proj.out
```

Example command lines for plotting the multibeam grid shown in Figure 9.1 follow.

```
grdgradient mb_grd -A270 -Nt -Ggrad_grd
grdimage mb_grd -Ctopo.cpt -Igrad_grd -Jm1 -K > Fig.9.1.ps
psbasemap -R-172.5/-167.5/-62/-60 -Jm1 -Ba2f1/a1:."Multibeam Bathymetry":WeSn
        -O -K >>Fig.9.1.ps
psscale -D2.5/-.5/2.5/.2h -Ctopo.cpt -B1000g1000f1000:meters: -I -N300 -O>>Fig.9.1.ps
```

The command lines for plotting the projected multibeam points shown in Figure 9.2 are below.

```
awk '{print $1,$2,$3}' mb.grav.proj.out | psxy -Ctopo.cpt -Sc.015 -R-93300/85700/-89800/115500
        -Jx.000028 -Ba50000f25000:meters::."Projected Multibeam":WeSn -K > Fig.9.2.ps
psscale -D2.5/-.5/2.5/.2h -Ctopo.cpt -B1000g1000f1000:meters: -N300 -O>>Fig.9.2.ps
```

There is more preparation needed for coherence analysis. The projected and rotated data points need to be gridded, and the grid needs to be "trimmed" so that almost all grid cells have values, while retaining the largest possible rectangular footprint. GMT routine "surface" is used to grid the data points- it fills in gaps in the data by interpolation, the "-R" option is used to select the desired area (within the trimming boundary, outlined in red in Figure 9.2), and the output is a grid with a 1 km grid spacing. The "surface" command line follows, and the grids, which are now ready for the coherence operation, are shown in Figure 9.3.

```
surface -R-84240/82120/-78790/81820 -I1000 mb.grav.proj.out -T0.25 -Gmb.surf.1km_grd
```

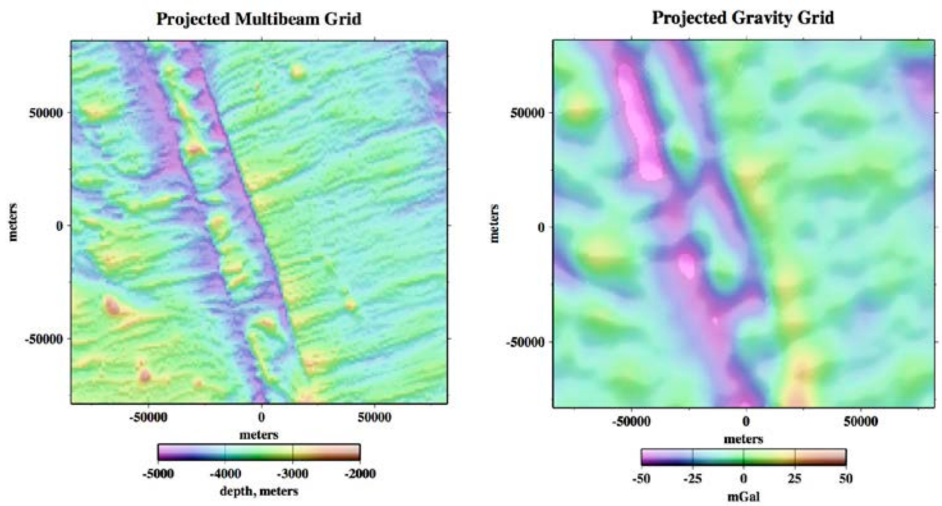


Figure 9.3 Multibeam (left) and gravity (right) grids fully prepared for coherence analysis.

The command lines to plot the multibeam grid shown in Figure 9.3 follow.

```
grdgradient mb.surf.1km_grd -A0 -Ne0.2 -Ggrad_grd
grdimage mb.surf.1km_grd -Igrad_grd -Ctopo.cpt -Jx.000028 -K>Fig.9.3.ps
psbasemap -R-84240/82120/-78790/81820 -Jx.000028 -Ba50000f25000:."Projected Multibeam
        Grid":WeSn -O -K>>Fig.9.3.ps
psscale –D2.329/-.5/2.5/.2h -Ctopo.cpt -B:"depth, meters": -I -N300 -O >>Fig.9.3.ps
```

9.1.4 Coherence Computation

We use GMT routine "gravfft" to compute the coherence between grids. This routine is a generalization of GMT routine "grdfft" which detrends the grids, tapers the edges, applies a two-dimensional Fast Fourier Transform (FFT), performs the coherence operation, and outputs the coherence averaged azimuthally as a function of wavelength.

The coherence results can be sensitive to the size of the input grid relative to the FFT grid dimensions used by "gravfft" for computation. The routine default selects dimensions that are larger than the input grid file size and that optimize the speed and accuracy of the FFT. Our experience has shown that the most reliable coherence results are obtained when the data grid is square and its dimensions are slightly smaller than the optimal FFT grid dimensions. The FFT dimensions can be specified with the "-N" option. Option "-Ns" prints out a table of suitable FFT dimensions, we list below sizes through 1024.

64, 72, 75, 80, 81, 90, 96, 100, 108, 120, 125, 128, 135, 144, 150, 160, 162, 180, 192, 200, 216, 225, 240, 243, 250, 256, 270, 288, 300, 320, 324, 360, 375, 384, 400, 405, 432, 450, 480, 486, 500, 512, 540, 576, 600, 625, 640, 648, 675, 720, 729, 750, 768, 800, 810, 864, 900, 960, 972, 1000, 1024

The command line for computing coherence with "gravfft" follows. Two grids are input- multibeam and gravity grids- and both have been prepared as described above. The output columns are wavelength in meters, coherence, and one sigma error.

```
gravfft mb.surf.1km_grd -Igrav.surf.1km_grd/wc > coh.out
```

Coherence for the Pacific-Antarctic spreading ridge study area is shown in Figure 9.4.

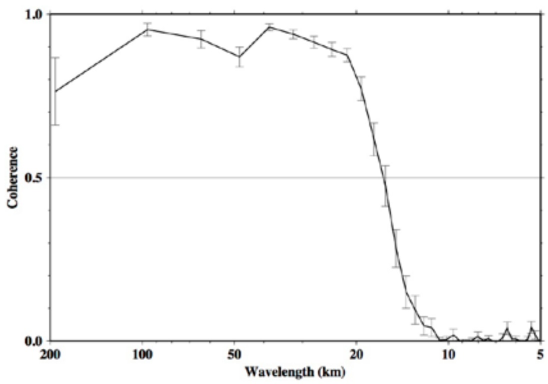


Figure 9.4 Radial coherence (black line) between grids shown in Figure 9.3. One sigma error bars are plotted.

For this area, bathymetry and gravity are coherent down to ~16 km.

The command lines used to plot Figure 9.4 follow.

```
awk ' {print $1/1000.,$2}' coh.out | psxy -R5/200/0/1 -JX-6l/4 -W4
        -Ba2f3:"Wavelength (km)":/a.5f.1g.5:"Coherence":WeSn -W4 -K > Fig.9.4.ps
awk '{print $1/1000.,$2,$3}' coh.out | psxy -Ey -R -JX -W4 -O >> Fig.9.4.ps
```

9.1.5 Interpreting Coherence Results

Figure 9.4 shows that gravity and bathymetry anomalies in this study area are coherent in the intermediate wavelength band. Coherence appears to decrease at wavelengths greater than ~100 km because longer wavelength topography is Airy compensated and the gravity signal is cancelled out. At shorter wavelengths, the coherence is low due to upward continuation of gravity from the seafloor to the sea surface. The gravity anomalies in Figure 9.1 (and Figure 9.3) look like a smoothed version of the multibeam bathymetry anomalies because of the upward continuation.

REFERENCES

Bendat JS Piersol AG (1986) Random data: Analysis and measurement procedures, 2nd edn., Wiley, New York

Sandwell DT, Smith, WHF (1997) Marine gravity anomaly from Geosat and ERS 1 satellite altimetry, J Geophys Res 102:10039-10054, doi:10.1029/96JB03223

ADVANCED TOPICS

Chapter 10.0 Uncertainty

10.1 Sources of Uncertainty

The following has been extracted from 5th Edition, S-44 (2008) Annex A, section A.4 by Rob Hare and editing for context

Although the following text focuses on uncertainties in data acquired with swath systems, it should be noted that it is in principle applicable to data acquired with any depth measurement system. A single-beam echosounder is just a special case of a multibeam echosounder (i.e. nadir beam only)

With swath systems the distance between the sounding on the seafloor and the positioning system antenna can be very large, especially in deep water. Because of this, sounding position *uncertainty* is a function of the errors in vessel heading, beam angle, refraction correction model and the water depth in addition to the uncertainty of the positioning system itself.

Roll and pitch errors will also contribute to the uncertainty in the positions of soundings. Overall, it may be very difficult to determine the position *uncertainty* for each sounding as a function of depth. The uncertainties are a function not only of the swath system but also of the location of, offsets to and accuracies of the auxiliary sensors.

The use of non-vertical beams introduces additional uncertainties caused by incorrect knowledge of the ship's orientation at the time of transmission and reception of sonar echoes. Uncertainties associated with the development of the position of an individual beam should include the following:

a) Positioning system uncertainty;
b) Range and beam angle uncertainty;
c) The uncertainty associated with the ray path model (including the sound speed profile), and the beam pointing angle;
d) The uncertainty in vessel heading;
e) System pointing uncertainty resulting from transducer misalignment;
f) Sensor location;
g) Vessel motion sensor errors i.e. roll and pitch;
h) Sensor position offset uncertainty; and
i) Time synchronization / latency.

Contributing factors to the vertical *uncertainty* include:

a) Vertical datum uncertainty;
b) Vertical positioning system uncertainty (if relevant);
c) Tidal measurement or prediction uncertainty, including co-tidal uncertainty where appropriate;
d) Range and beam angle uncertainty;

e) The uncertainty associated with the ray path model (including the sound speed profile), and the beam pointing angle;
f) Ellipsoidal / vertical datum separation model uncertainty (if relevant);
g) Vessel motion uncertainty, i.e. roll, pitch and heave;
h) Vessel dynamic draught, including static draft, settlement and squat;
i) Seabed slope (when combined with positioning uncertainty); and
j) Time synchronization / latency.

Agencies responsible for the survey quality are encouraged to provide uncertainty budgets for their own systems.

10.2 Measuring uncertainty

10.2.1 Methods of quality assessment, artifacts

TEXT IN PREPARATION

10.2.2 Comparisons with multibeam

TEXT IN PREPARATION

Local

TEXT IN PREPARATION

Regional

TEXT IN PREPARATION

Global

TEXT IN PREPARATION

10.2.3 Monte Carlo Technique

Contributed by Paul Elmore, Naval Research Laboratory, Stennis Space Center, USA

A peer-reviewed methodology for estimating error on historic data sets using a Monte Carlo technique is published in Jakobsson et al. (Jakobsson et al. 2002); Figure 10.1 illustrates the procedure. It can be used on either historic or new soundings to provide an error layer for the GMT Splines-In-Tension routine (Surface routine). In order for this technique to work, estimates of the horizontal and vertical uncertainties for the soundings are needed.

In the simulations, the original two-dimensional navigation positions and the one-dimensional soundings are randomly perturbed using a normally distributed random number generator (RNG). Let the number of original soundings be J, the number of surveys be K, the horizontal navigation uncertainty of the K^{th} survey be H_K, the vertical sonar uncertainty of the K^{th} survey be V_K, the number of output grid points be I, and the number of Monte Carlo simulations be N, with each loop denoted by n. For the K^{th} survey, the RNG perturbs the position data ~ $\mathcal{N}(0, H_K^2)$[†] and the vertical position ~ $\mathcal{N}(0, V_K^2)$. The gridded bathymetry surface is constructed from the GMT Surface (or other interpolator of choice) during each loop of the Monte Carlo simulation, resulting in N different interpolated bathymetry surfaces. The gridded uncertainty estimate is then the standard deviation of the N surfaces at each i^{th} grid point, $i \in I$. Assuming that all sounding used have been cleaned and corrected for ship dynamics, sound speed profile errors, etc., positional uncertainty and the bottom slope predominantly influence the bathymetric uncertainty estimated from this method.

A benefit of this method is that it is relatively simple to code: one simply needs the interpolation and RNG routines called within a Monte Carlo shell that perturbs the horizontal and vertical positions. This procedure, however, is computationally intensive and requires the use of original soundings data. Attempting this approach for a large number of soundings data may not be pragmatic.

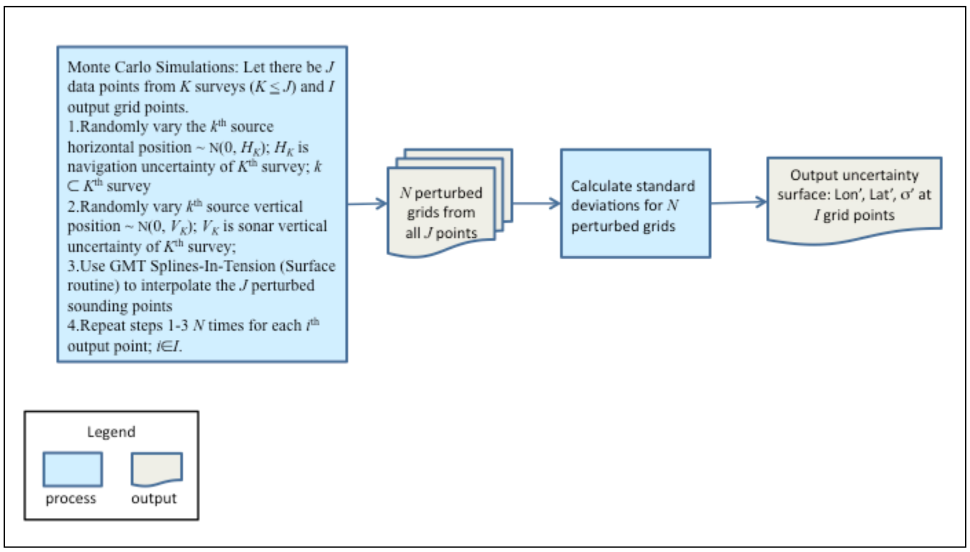


Figure 10.1 **Process flow for the Monte Carlo procedure of Jakobsson et al (2003).**

10.2.4 Bayesian Network Technique

Contributed by Paul Elmore, Naval Research Laboratory, Stennis Space Center

---

[†] *The notation ~ $\mathcal{N}(\mu, \sigma^2)$ means that the quantity follows a normal, or Gaussian, probability distribution with mean μ and variance σ^2*

When the data sets are large, a Bayesian network (BN; a good discussion is given in Chapter 14 of Russell and Norvig (2003)) can be an alternative to the Monte Carlo method. With this approach, published in Elmore et al. (2009) and illustrated in Figure 10.2, causal relationships of navigation error and bottom slope to bathymetric uncertainty are quantified by conditional probability densities (CPD's). One uses the Monte Carlo technique on representative sets of soundings data to tabulate the CPD's necessary for the statistical inference. The BN then produces a histogram of this uncertainty estimate for an area given the navigation errors used to survey the region and bottoms slopes that are present. An intermediate uncertainty is obtained from the mean plus one standard deviation of the histogram. The final uncertainty estimate is obtained by first adding the square of this intermediate uncertainty to the square of the vertical error estimate under the assumption of statistical independence between the two, then computing the square root of this sum. This type of computation is much less costly than the Monte Carlo technique both in computation overhead and required input data, potentially leading to a significantly large (two orders of magnitude) increase in computational speed. Estimates of the horizontal and vertical uncertainty for the soundings must be available to both train the network and run the Bayesian Network.

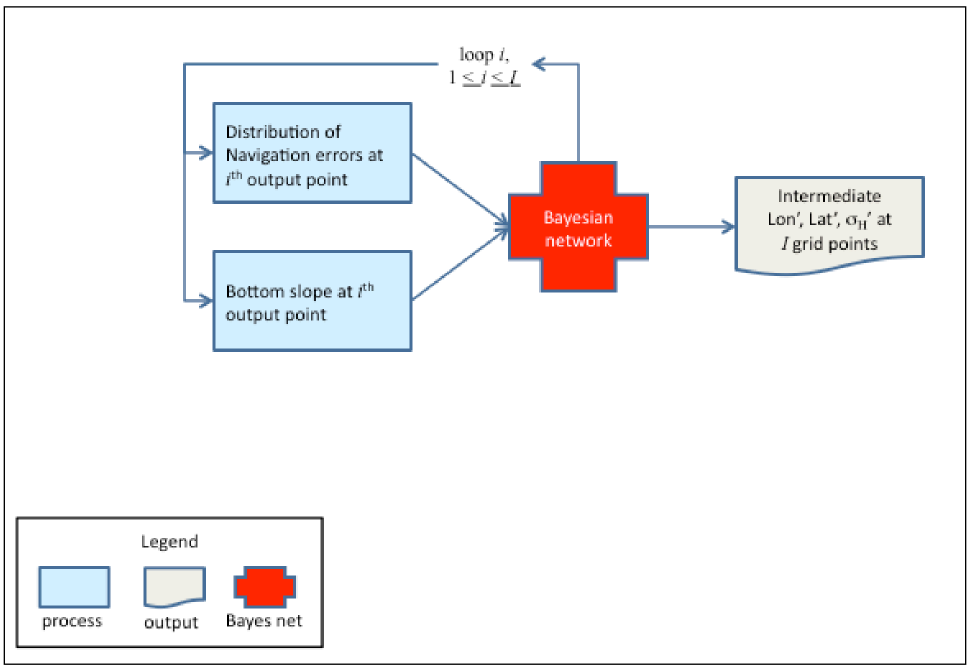


Figure 10.2 **Conceptual flow of the Bayesian Network approach.**

Bayesian Network Design and Access

The Bayesian network itself can be programmed using commercial Netica software, Version 4.0.8 (2008), which provides a GUI for construction and programming of the BN and (as of this writing) is free to use for the small Bayesian network created here. Figure 10.3 shows the network topology as displayed by the Netica GUI from Elmore et al (2009). It consists simply of two main parent nodes, one for navigation error and one bottom slope, and one child node, the intermediate uncertainty estimate. In this example, there are eight different surface navigation uncertainties categories; Table 10.1 associates historical navigation techniques to uncertainty. The bottom slope and bathymetric uncertainties follow a logarithmic binning scheme so that one significant digit is maintained in the uncertainty estimate while keeping the BN to a manageable size. Once created, the software stores the BN as either a binary object or ASCII file that can be accessed and manipulated through the use of vendor provided application program interface (API) libraries.

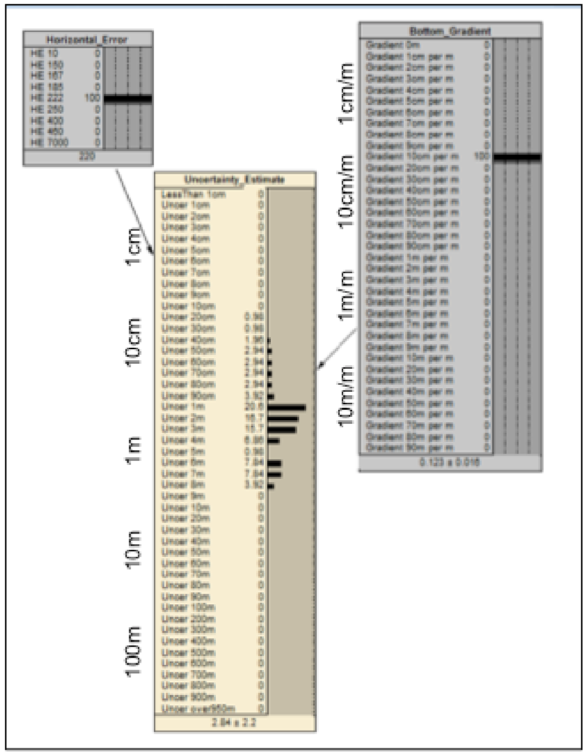


Figure 10.3 **Bayesian network topology as displayed by the Netica software GUI. The "Horizontal_Error" node variables follow the navigation errors provided in Table 1, with the numbers for surface navigation error in meters. The "Bottom_Gradient" and "Uncertainty_ Estimate" nodes follow a logarithmic binning scheme.**

Table I: Horizontal Error Categories

Navigation Mode	Accuracy	Navigation Mode	Accuracy
GPS/SINS (3 or more Satellites)	10-15 m	NAVSAT/Single Range LORAN/SINS	250 m
GPS/DR (3 or more Satellites)	10-15 m	NAVSAT/ SINS	250 m
NAVSAT/Range Range LORAN/SINS	150 m	NAVSAT/Single Range LORAN/DR	250 m
NAVSAT/Range Range LORAN/DR	167 m	NAVSAT/DR	400 m
NAVSAT/Hyperbolic LORAN/SINS	185 m	LORAN/SINS LORAN/DR	463 m
NAVSAT/Hyperbolic LORAN/DR	222 m	Satellite Altimetry	7000 m

Table 10.1 Navigation uncertainty categories used in Elmore et al. (2009) for the Bayesian Network approach to uncertainty estimation. The abbreviations are as follows: GPS – Global Positioning System; SINS – Ship's Inertial Navigation System; DR – Dead Reckoning; NAVSAT – Navy Navigation Satellite System; LORAN – Long Range Navigation. The "/" means that two or more techniques are combined.

Bayesian Network Training Procedures

Using the procedure from Jakobbson et al. (2003), one ascertains the propagation of navigational error and bottom slope into bathymetric uncertainty by Monte Carlo simulations. Let there be a total on M navigation error categories. The results of the simulations (a total of $M \times N$ simulations) to populate the CPT of the BN, or "train" the BN. Figure 10.4 shows a high level view of the entire training process, bracketed into three main blocks: 1) Monte Carlo simulations, 2) standard deviation and slope calculations, and 3) Bayes network programming.

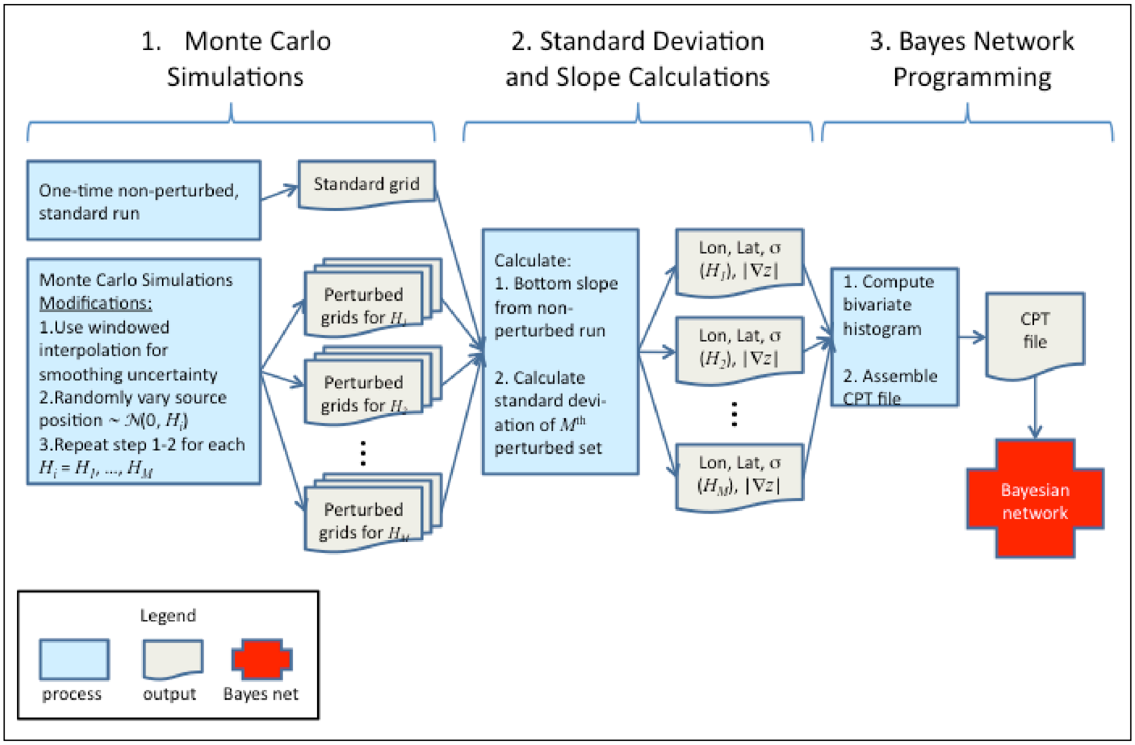


Figure 10.4 High level view of the Bayesian network training process. The figure illustrates computation of the standard deviations as modified from Jakobsson (2003), and use of these results to construct the conditional probability tables (CPT) for the Bayesian network.

Figure 10.5 shows a low level flow chart for the Monte Carlo block. In the Monte Carlo simulations, the original locations of the sounding points (or gridded points acting as sounding points) are first moved in random directions. A normal RNG (Marsaglia and Tsang 2000) provides deviations for the north and east positions as specified by the horizontal error category for the m^{th} simulation loop, H_m ($1 \leq m \leq M$), such that the RNG ~ $\mathcal{N}(0, H_m^2)$. The data points are then interpolated to provide a bathymetry surface and stored to an ASCII text file, with the Monte Carlo loop number and navigation error stored in the file name. As in Jakobbson et al. (2003), N = 100 Monte Carlo iterations may be performed for each m^{th} set of simulations; however, a difference between the two is that the set of N Monte Carlo simulations are then repeated M times for each horizontal error category in order to fully populate the CPT of the BN. Another difference from the Monte Carlo procedure is that each sounding is assumed to have the same probability distribution of horizontal errors so that one can obtain a statistical distribution for bathymetric uncertainty for a range of horizontal errors. Although the actual data do not necessarily have these errors, it is assumed they do in order to train the network. Also, the vertical positional will be treated algebraically as discussed in Section 3 below.

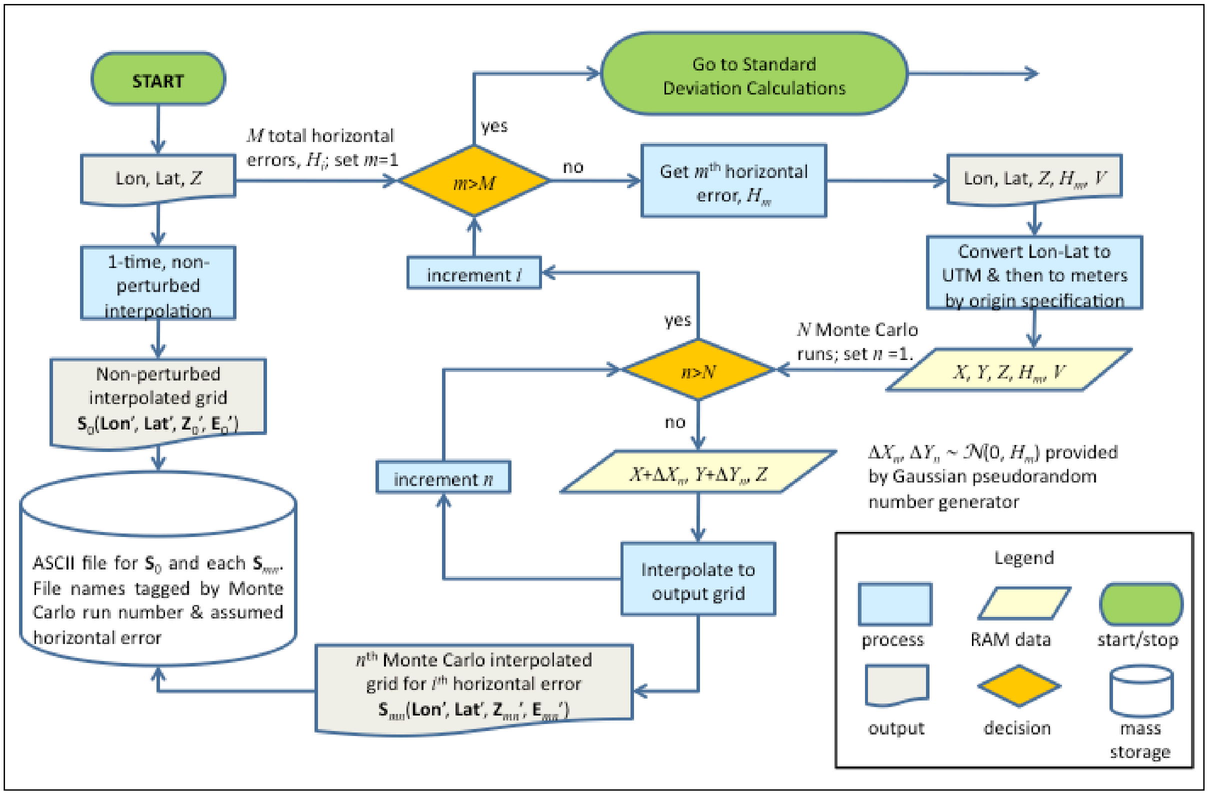


Figure 10.5 **Low level flow chart of the Monte Carlo simulations block for the Bayesian network training process. Here, *X* and *Y* are the east and north positions of the output grid points in meters after conversion to Universal Transverse Mercator (UTM) projection and specification of a UTM origin in the grid, *Z* is interpolated depth, S_o is an unperturbed interpolation, S_{mn} is the perturbed grid for the n^{th} Monte Carlo loop and m^{th} Horizontal Error Category, E' is error from smoothing. The bold face means that the quantity is a vector.**

Figure 10.6 shows a low level flow chart for the next block of the training process, the standard deviation and slope calculations block. After completion of the final iteration, the uncertainty estimate for the set of *I* output points are computed from the gridded standard deviation of the Monte Carlo bathymetry surfaces and stored to ASCII files for use below (low level flow chart shown in Fig. 10.7). Specifically, the ASCII files contain longitude, latitude, bathymetry, Monte Carlo standard deviation and magnitude of the two-dimensional slope. To obtain the slopes, an unperturbed calculation of the bathymetry surface is first made, followed by computation of the bottom gradient throughout the grid. One can use finite differences as discussed in Zhou and Liu (2004) and Oksanen and Sarjakoski (2005) to calculate bottom gradient. Longitude and latitude positions are translated to UTM coordinates so that positional differences and gradients can be computed in meters. Linear extrapolation is used to aid computation of the gradient along the computational edges.

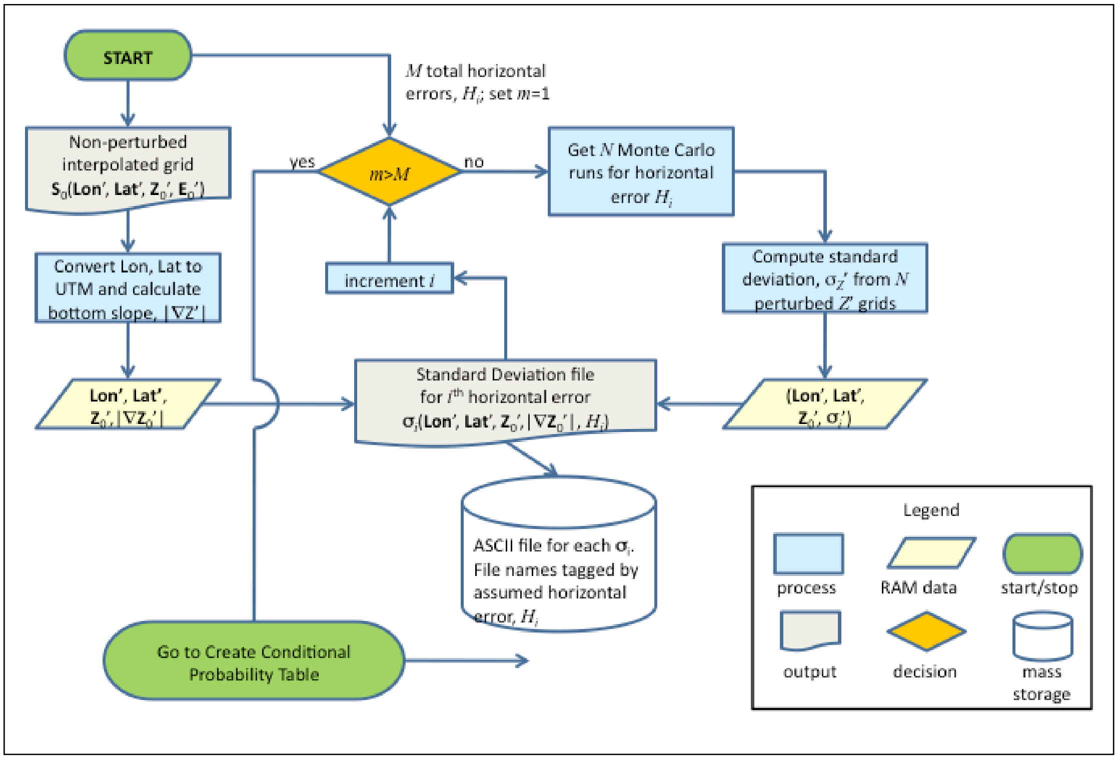


Figure 10.6 **Low level flow chart of the Standard Deviations block for the Bayes net training process.**

Figure 10.7 shows a low level flow chart for the final block of the training process, the Bayes network programming block. The BN is programmed using the graphic user interface (GUI) in Netica on a Windows based PC. The ASCII files containing the Monte Carlo uncertainties and bathymetric slopes are first read from the standard deviation file discussed in the above section for the first navigation uncertainty category. Then, using the logarithmic binning scheme discussed above we compute the bivariate histogram (i.e., three-dimensional histogram) with slopes and bathymetry uncertainty, respectively, indexing the row and column bins of the bivariate histogram. This histogram is normalized and written to an ASCII file. This process is then repeated for subsequent navigation uncertainties with the resultant normalized bivariate histograms appended to the same output file.

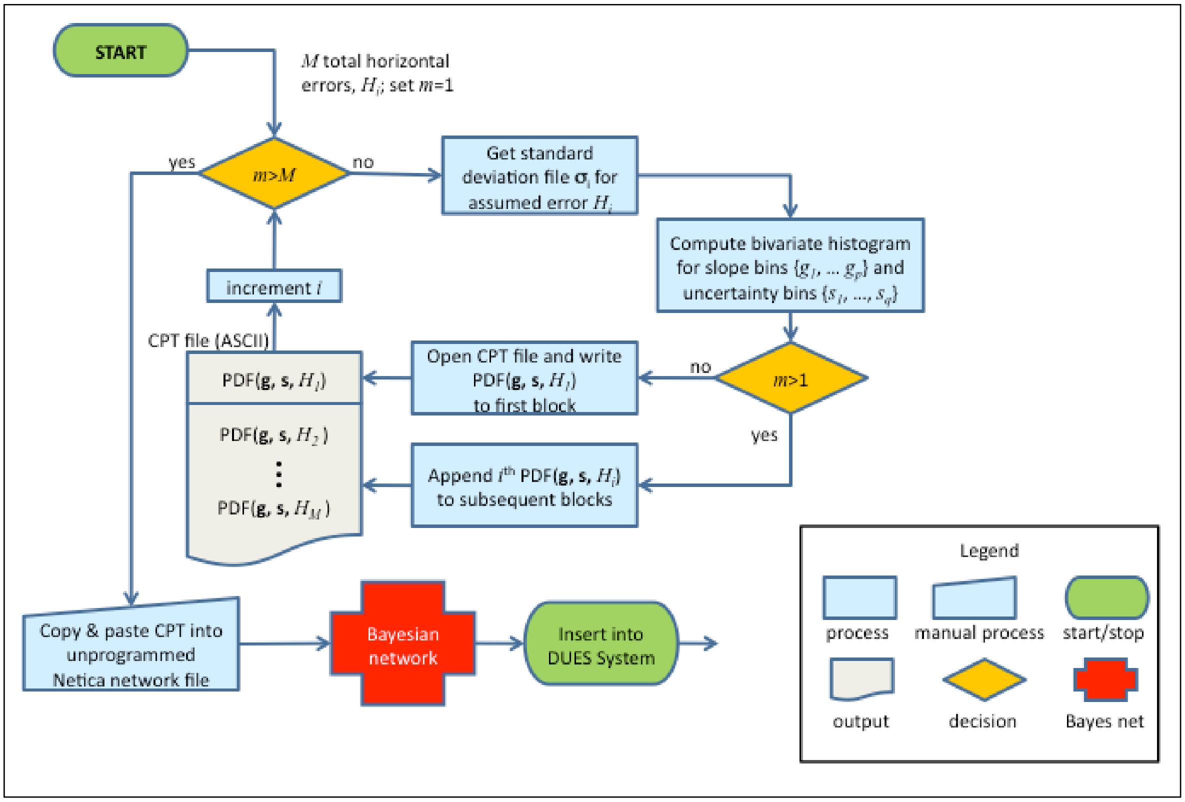


Figure 10.7 Low level flow chart of the Bayes network programming block for the Bayesian network training process. The network is stored in a library file that is accessed by API.

After completing this process, the Bayesian network *.neta file is opened in the Netica GUI, and the CPT for the "Uncertainty Estimate" node is displayed. The CPT is programmed by simple "copy-and-paste" from the ASCII file into the spreadsheet-like data display for the CPT in the GUI. The CPT is already arranged to have the same stacked structure as the ASCII file.

Operational Flow

Figure 10.8 shows how the BN could be integrated into the overall operational system, with key components and data flow. The end user of the system first defines a region of interest for uncertainty estimation, followed by query of available navigation and bathymetry for the area. The navigation metadata would indicate survey era and platform used from the navigation data to obtain the types of navigation (for horizontal uncertainty) and sonar systems (for vertical uncertainty) used for the track lines in the area. The fractional lengths of track lines with like uncertainties could be calculated (i.e. sum of all track lengths with a like error divided by the total of all track lengths) and become the weights for the BN's Horizontal Error node. Also extracted is the bathymetry from the same area to obtain the bottom gradient. The system loops through each grid point, assigns the bottom gradient to the appropriate bin of the BN's Bottom Gradient node, and extracts the resultant histogram. Calculation of mean plus one standard deviation for this histogram provides the intermediate uncertainty. The final uncertainty estimate is obtained by first adding the square of the intermediate uncertainty to the square of the vertical uncertainty estimate.

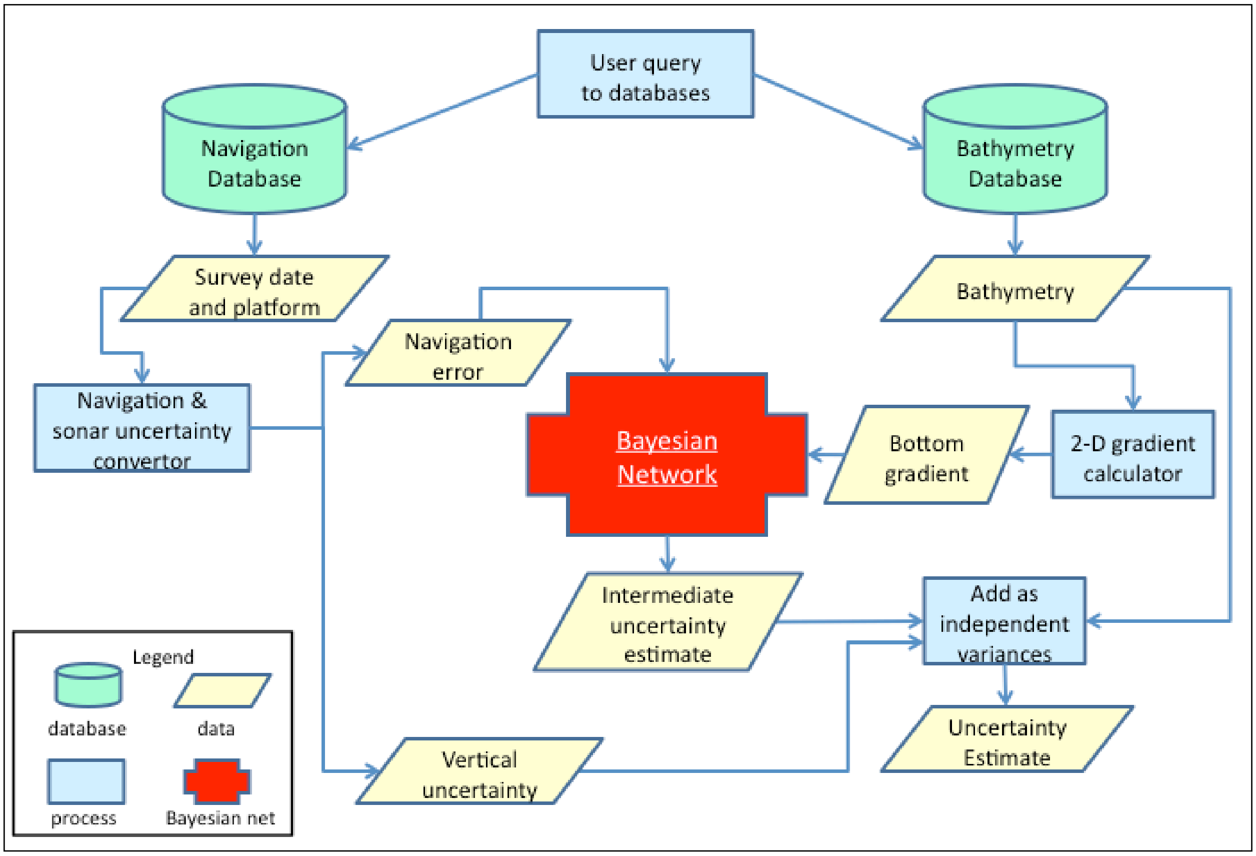


Figure 10.8 Integrated flow chart of the DBDB-V Uncertainty Expert System (DUES) after programming of the Bayes network component is complete.

10.2.5 Windowed Regression with Kriging Corrections

Contributed by Paul Elmore, Naval Research Laboratory, Stennis Space Center

As an alternative to using GMT surface, localized regression techniques of Cleveland (1979) have been applied to bathymetry problems in Plant et al. (2002). This technique was then refined by Calder (2006) to provide refinements to the interpolated surface and uncertainty estimate using ordinary kriging and the Monte Carlo technique discussed in Section A above. An advantage of this technique is that it provides an uncertainty estimate with the interpolated surface. A disadvantage is that it requires dense data sounding to be meaningful. The discussion that follows below closely to the discussions in the papers by Cleveland (1979) and Calder (2006).

Linear Smoothing by Locally Weighted Regression (Loess)

Linear smoothing techniques construct the interpolated surface, $\hat{z}(\mathbf{s})$, $\mathbf{s} \equiv$ matrix of (x, y) coordinates, from a linear weighted average of known data values. Unlike the splines-in-tension technique that finds a global solution to the available data, this technique obtains an interpolated value at a grid point by using only a subset of neighboring points. Mathematically,

$$\hat{z}(\mathbf{s}) = \sum_i \alpha(\mathbf{s} - \mathbf{s}_i) z(\mathbf{s}_i), \tag{(i)}$$

where $\alpha(\mathbf{s} - \mathbf{s}_i)$ specifies the smoother coefficients and the index i corresponds to the subset of local points, $z(\mathbf{s}_i)$, to be used for the interpolated value at $\mathbf{s}$, $\hat{z}(\mathbf{s})$. A common technique used to specify these coefficients is locally weighted regression (or "loess"), first published in Cleveland (1979) and further developed in Cleveland and Devlin (1988). A textbook by Givens and Hoeting (2005) also discusses the technique. The methodology determines the smoother coefficients from a weighted least squares polynomial (linear or quadratic) fit of windowed data.

1. Methodology

To summarize Cleveland's methodology, the two-dimensional case is considered first. These equations to the three-dimensional case in Section B, but the methodology will remain the same. Let data points x_i and y_i be related as

$$y_i = g(x_i) + \varepsilon_i, \tag{(ii)}$$

where $g(x)$ is a smooth function and ε_i is Gaussian noise with zero mean and variance σ^2. Define $\hat{y}_i$ to be the estimate of $g(x_i)$ (i.e. $\hat{y}_i \approx g(x_i)$). Let

- the number n be the predetermined number of data points to be used for estimating $\hat{y}_i$
- the set x_k, $k = 1,\ldots,n$, be the subset of x_j's ($j = 1,\ldots,N$; N = total number of data points, $n < N$) that are closest to x_i
- the distance h_i be the distance from x_i to the furthest x_k.
- the windowing weights to be used for the regression, $w_k(x_i) = W([x_k - x_i]/h_i)$, where $W(x)$ is the tricube function, defined as

$$W(x) \equiv \begin{cases} \left(1 - |x|^3\right)^3, & x < 1 \\ 0 & , \text{otherwise} \end{cases} \tag{(iii)}$$

With these definitions, the loess procedure calculates the set of polynomial coefficients, $\hat{\beta}_l(x_i)$, that are the values for the β_l's that minimize

$$q(x_i) = \sum_{k=1}^{n} w_k(x_i)\left(y_k - \beta_0 - \beta_1 x_k - \cdots - \beta_d x_k^d\right)^2, \tag{(iv)}$$

where $q(x_i)$ is the error function. The interpolated value for $g(x_i)$ is then

$$\hat{y}_i = \sum_{l=0}^{d} \hat{\beta}_l (x_i) x_i^d = \sum_{k=1}^{n} \alpha_k (x_i) y_k \; . \qquad ((v)$$

Some points of interest are as follows (Cleveland 1979).

- The coefficients obtained have a decreasing trend to the edge of the window so that the data centered in the window generally have the greatest influence.
- The size of the polynomial typically used in Eqn. (((iv) is d = 1 or 2. For d = 0, the result is a simple moving average. The d = 1 case is called linear loess or "lowess" smoothing, and d= 2 is quadratic "loess" smoothing.
- Cubic and higher fits typically are not used as the fits can become over fitted and numerically unstable.
- This technique also has a robustness option, so that the interpolation can be shielded from the effects of outliers in the data.
- Cleveland choose the tricube weighting function because it allowed the estimate of the error variance to be approximated by a chi-square distribution and usually lowered the variance of the estimate surface as the number of points used for the estimate increased.

Errors propagated into the interpolation by the technique are straight forward to compute. Under the assumption the data follow Eqn. (((ii), the estimate of the variance for $\hat{y}_i$, $\hat{\sigma}_i^2$, is (Plant et al. 2002)

$$\hat{\sigma}_i^2 = \sigma^2 \sum_{k=1}^{n} \left[\alpha_k (x_i) \right]^2 , \qquad ((vi)$$

which is derivable from independent error propagation.

2. Convolution Approach

Figure 10.9 plots the set of α_k's as determined for a centered impulse response using the linear and quadratic loess interpolators. These coefficients are the smoother weights in Eqn. (((v) and are also called the "equivalent kernel". They depend only on the grid points, and, due to their finite width, act as a window or low-pass filter function on the spatial data (i.e. the smoothing weights and the data undergo convolution). Hence, one could bypass solving a weighted least square problem and simply compute the convolution of the smoothing window with the data, which should be computationally faster. In addition, other windowing functions could be used to perform the smoothing.

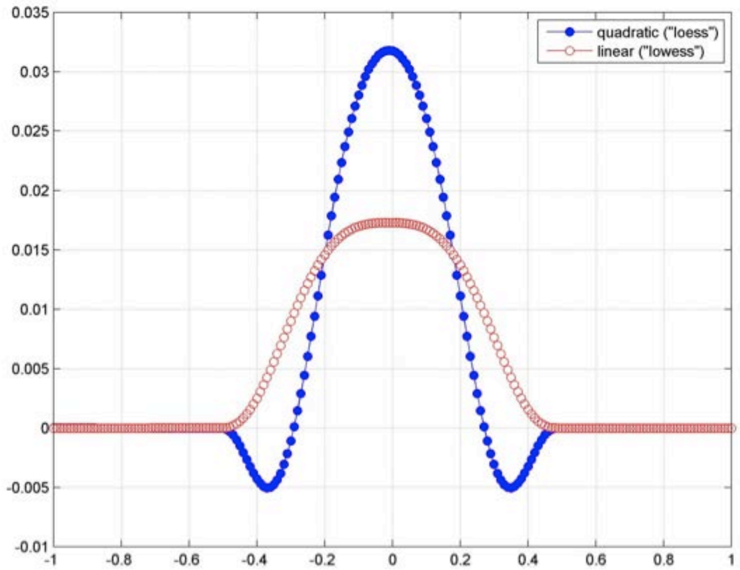


Figure 10.9 **Equivalent kernel weights for linear and quadratic loess windows.**

The use of other weighting functions would also allow one to interpolate over data points with differing accuracy. In this case, Eqns. (((v) and (((vi) become

$$\hat{y}_i = \sum_{k=1}^{n} \alpha_k(x_i)\left(\sigma_k^{-2} \Big/ \sum_{k=1}^{n} \sigma_k^{-2}\right) y_k = \sum_{k=1}^{n} \tilde{\alpha}_k(x_i) y_k\,, \qquad ((\text{vii})$$

$$\hat{\sigma}_i^2 = \sum_{k=1}^{n} \sigma_k^2 \tilde{\alpha}_k^2(x_i) \qquad ((\text{viii})$$

where σ_k^2 is the variance of y_k and $\tilde{\alpha}_k(x_i) \equiv \alpha_k(x_i)\left(\sigma_k^{-2} \Big/ \sum_{k=1}^{n} \sigma_k^{-2}\right)$.

Combining windowing with ordinary kriging corrections

Since the above regression technique smooths the data, Calder (2006) provides an ordinary kriging step for correcting at the sounding points with the Monte Carlo technique to accounts for navigation uncertainty. The summary of the complete methodology is as follows.

1. Following Plant et al. (2002), interpolate the data with the quadratic loess interpolation technique of Cleveland (1979) to provide a trend surface for the bathymetry and uncertainty layer.
2. Restore finer details smoothed by the interpolation from ordinary kriging of the residuals; add the errors associated with ordinary kriging to the uncertainty layer from Step 1

assuming statistical independence (i.e. the variances add). The surface generated from kriging the residuals is the residual surface.

3. Estimate additional uncertainty caused by positional errors from the Monte Carlo technique of Jakobsson et al. (2002), but repeat steps 1 and 2 above instead of using the splines-in-tension algorithm at each iteration. As before, add the estimated error to the uncertainty layer by assuming statistical independence.

In equation form, the final bathymetry surface, $Z(\mathbf{s})$, is the sum of the trend surface, $\mu(\mathbf{s})$, and the residual surface, $R(\mathbf{s})$.

$$Z(\mathbf{s}) = \mu(\mathbf{s}) + R(\mathbf{s}) \tag{(ix}$$

The uncertainty layer, $\sigma_Z(\mathbf{s})$, is

$$\sigma_Z(\mathbf{s}) = \left[\sigma_\mu^2(\mathbf{s}) + \sigma_R^2(\mathbf{s}) + \varepsilon^2(\mathbf{s})\right]^{1/2} \ , \tag{(x}$$

where $\sigma_\mu(\mathbf{s})$ and $\sigma_R(\mathbf{s})$ are the uncertainties for the trend and residual layers, respectively, and $\varepsilon(\mathbf{s})$ is the uncertainty layer associated with positional errors.

1. Quadratic Loess Interpolation for Trend Surface

Extending Eqn. (((iii) through (((vi) to two dimensions, the equations for calculating the trend surface, $\mu(\mathbf{s})$ in Eqn. (((ix) are as follows.

$$\mu(\mathbf{s}) = \mathbf{p}^T(\mathbf{s})\hat{\boldsymbol{\beta}}(\mathbf{s}), \tag{(xi}$$

where $\mathbf{p}^T(\mathbf{s}) \equiv [x^2, y^2, xy, x, y, 1]$ and vector $\hat{\boldsymbol{\beta}}(\mathbf{s}) \equiv [\hat{\beta}_5(\mathbf{s}), \ldots, \hat{\beta}_0(\mathbf{s})]^T$ being the set of $\beta_n(\mathbf{s})$'s that minimize the weighted least squares

$$q(\mathbf{s}) = \sum_{k=1}^{n} w_k(\mathbf{s})\left(z_k - \beta_0(\mathbf{s}) - \beta_1(\mathbf{s})y_k - \beta_2(\mathbf{s})x_k - \beta_3(\mathbf{s})x_k y_k - \beta_4(\mathbf{s})y_k^2 - \beta_5(\mathbf{s})x_k^2\right)^2 \tag{(xii}$$

using the two-dimensional tri-cube weighting function

$$w_k(\mathbf{s}) = \begin{cases} \left(1 - \left|\dfrac{\mathbf{s} - \mathbf{s}_k}{d_0}\right|^3\right)^3, & \left|\dfrac{\mathbf{s} - \mathbf{s}_k}{d_0}\right| < 1 \\ 0 & , \text{ otherwise} \end{cases}, \tag{(xiii}$$

and the user provided value for d_0, which Calder's paper suggests that d_0 be ten times the largest sample spacing. Information lost by oversmoothing is regained when the residuals are calculated in the kriging portion.

Estimates for the variances follow Eqn. (((vi) for like variances in all the data or $\hat{\sigma}^2(\mathbf{s}) = \sum_{k=1}^{n}\left[\alpha_k(\mathbf{s})\sigma_k\right]^2$ for differing variances. The $\alpha_k(\mathbf{s})$'s now have azimuthal symmetry and the same radial dependence as the 1-D case.

In addition, the estimate is made more robust (i.e. eliminate outliers) by flagging estimates greater than three Mahalanobis units as "no data". The Mahalanobis distance (Mahalanobis 1936), $\mathbf{M}(\mathbf{p},\mathbf{m},\mathbf{C})$, is

$$\mathbf{M}(\boldsymbol{\mu},\mathbf{m},\mathbf{C}) \equiv (\boldsymbol{\mu}-\mathbf{m})^T \mathbf{C}^{-1}(\boldsymbol{\mu}-\mathbf{m}) \qquad ((\text{xiv})$$

where $\boldsymbol{\mu}(\mathbf{s})$ is the vector of all estimated depths and $\mathbf{m}(\mathbf{s})$ and $\mathbf{C}(\mathbf{s})$ are the corresponding mean depth measurements and covariances for the windowed (c.f. Eqn. (((xiii)) data sets. To intuitively explain Eqn. (((xiv), suppose that the data points are all independent and have mean m and variance σ^2. The $\mathbf{C}$ matrix is all zeros except for the diagonal elements, which are all σ^2. Then, if an estimate differs from its corresponding windowed mean by more than three variances (i.e. $(\mu - m)^2/\sigma^2 > 3$), that estimate receives the "no data" mark. Eqn. (((xiv) generalizes this simpler scenario to account for covariance between the data.

2. Kriging the Residuals

Since loess interpolation is not exact at the data points, residuals, $R(\mathbf{s}_i)$, exists between actual measurements, $z(\mathbf{s}_i)$, and the corresponding estimated depths along trend surface, $\mu(\mathbf{s}_i)$, such that $R(\mathbf{s}_i) = z(\mathbf{s}_i) - \mu(\mathbf{s}_i)$. The residual surface in Eqn. (((ix) is found from interpolation of the $R(\mathbf{s}_i)$ set using ordinary kriging, as an overall but constant unknown bias may exist in the residuals. The variogram will likely have directional anisotropy, so that $2\gamma(\mathbf{s}_i, \mathbf{s}_j) = 2\gamma(h, \theta)$, where h is the separation distance between the two points and θ is the heading angle (clockwise from the north).

To account for this anisotropy, the following semivariance, $\gamma_D(\mathbf{s}_i, \mathbf{s}_j)$, is used and constructed (unconditionally valid for two-dimensions (Brandt 1998)) in the following manner:

1) Obtain an empirical estimate of the variogram
2) Compute the average azimuthal variogram from the empirical estimate to detect the directions of minimum and maximum variation,
3) Fit the variograms along these two axes to the spherical model for variograms
4) Using parameters from the fits to the spherical model and the direction for minimum variance, construct $\gamma_D(\mathbf{s}_i, \mathbf{s}_j)$ from Eqns. (((xx) – (((xxiii) below. These steps are now discussed in more detail.

Step 1: Following page 69 in Cressie (1993), the variogram for the residuals is approximated using the methods-of-moments (or classical) estimator in blocks sizes = $2d_0$ (c.f. Eqn. (((xiii)), so that

$$2\gamma(h,\theta) \approx 2\hat{\gamma}(h_i,\theta_i) = \frac{1}{|N(h_i)||N(\theta_j)|} \sum_{(a,b)\in N(h_i) \cap N(\theta_j)} \left(R(\mathbf{s}_a) - R(\mathbf{s}_b)\right)^2 . \tag{(xv)}$$

The sets $N(h_i)$ and $N(\theta_i)$ contain binned separation distances and heading angles as defined by the equations

$$N(h_i) \equiv \left\{(a,b): h_i - \frac{\Delta h_i}{2} \le d(\mathbf{s}_a, \mathbf{s}_b) \le h_i + \frac{\Delta h_i}{2}\right\} \tag{(xvi)}$$

$$N(\theta_j) \equiv \left\{(a,b): \theta_j - \frac{\Delta \theta}{2} \le \angle(\mathbf{s}_a, \mathbf{s}_b) \le \theta_j + \frac{\Delta \theta}{2}\right\}, \tag{(xvii)}$$

where $d(\mathbf{s}_a, \mathbf{s}_b)$ and $\angle(\mathbf{s}_a, \mathbf{s}_b)$ are the Euclidian distance and heading angle between $\mathbf{s}_a$ and $\mathbf{s}_b$ and $|N(h_i)|$ and $|N(\theta_i)|$ are the number of bins in these sets. For the data set analyzed in Calder (2006), $\Delta h_i = 0.05 d_0$, $\Delta\theta_i = 45^{\circ}$, $h_i = i\Delta h + \Delta h/2$ and $\theta_j = j\Delta\theta$.

Step 2: The average azimuthal variogram, $2\bar{\gamma}(\theta_j) \equiv |N(h_i)|^{-1} \sum_i 2\hat{\gamma}(h_i, \theta_j)$ is then calculated. In Calder (2006), $2\bar{\gamma}(\theta_j)$ had two sets of maximum and minima (caused by trending ridges), so it could be modeled by the function

$$2\bar{\gamma}(\theta_j) \approx 0.5 g_0 + g_2 \cos(4\pi\theta_j + \phi_2), \tag{(xviii)}$$

which is the second Fourier eigenfunction, Equation (((xviii) is fitted to the data by evaluating the second discrete Fourier transform coefficient. The phase constant, ϕ_2, is the radian angle (which goes counterclockwise from the east) where the first minimum is found. It is changed to a heading angle (again, clockwise from the north), θ_m, from the transformation $\theta_m \equiv -\phi_2/2 + \pi/2$. This angle is obtained for each $2d_0$ block, then interpolated to get $\theta_m(\mathbf{s}_i)$ for Step 4 below.

Step 3: Two finer-scaled directional variograms are then calculated; one in the θ_m direction, the other in the perpendicular direction, $\theta_m^{\perp}$; using the methods-of-moments estimator with angle bins of $\pm\pi/2$ about each direction. In this case, $\Delta h = 0.02 d_0$ and the data were truncated to those within 95% of the mean to reduce outliers. These empirical variograms are fitted to the standard spherical variogram model (Cressie (1993), Eqn. (2.3.8); Davis (2002), Eqn. (4.98))

$$2\gamma(h) = \begin{cases} 0, & h = 0 \\ a_0 + a_1\left(1.5(h/a_2) - 0.5(h/a_2)^3\right), & 0 \le h/a_2 \le 1 \\ a_0 + a_1, & h/a_2 \ge 1 \end{cases} \tag{(xix)}$$

using the Levenberg-Marquardt algorithm (Marquardt 1963; Brandt 1999) for fitting (i.e. solving for a_0, a_1 and a_2). Let the modeled variogram along the θ_m direction be $2\gamma_0(h)$ and the fitting constants be a_0, a_1 and a_2. Similarly, let the modeled variogram along the $\theta_m^{\perp}$ direction be $2\gamma_0^{\perp}(h)$, and the fitting constants be $a_0^{\perp}$, $a_1^{\perp}$, and $a_2^{\perp}$. When solving for both sets of constants, the variograms are constrained to equal the same sill at large distances so that $a_0^{\perp} = a_0$ and $a_1^{\perp} = a_1$, but $a_2^{\perp} \neq a_2$ in general.

Step 4: Define the anisotropy parameter, $\alpha_{aniso} \equiv a_2 / a_2^{\perp}$. Then, $\gamma_D(\mathbf{s}_i, \mathbf{s}_j)$, is evaluated in the following manner.

$$\gamma_D(\mathbf{s}_i, \mathbf{s}_j) = \gamma_0(d'(\mathbf{s}_i, \mathbf{s}_j)), \qquad \text{((xx)}$$

where

$$d'(\mathbf{s}_i, \mathbf{s}_j) \equiv \left\| \mathbf{A}(\theta_m(\mathbf{s}_i), \alpha_{aniso})(\mathbf{s}_i - \mathbf{s}_j) \right\|, \qquad \text{((xxi)}$$

$$\mathbf{A}(\theta_m(\mathbf{s}_i), \alpha_{aniso}) = \mathbf{R}(-\theta_m(\mathbf{s}_i))\, diag(1, \alpha_{aniso})\, \mathbf{R}(\theta_m(\mathbf{s}_i)) , \qquad \text{((xxii)}$$

and $\mathbf{R}(\theta_m(\mathbf{s}_i))$ is the standard two-dimensional rotational matrix (Goldstein 1980)

$$\mathbf{R}(\theta_m(\mathbf{s}_i)) = \begin{bmatrix} \cos(\theta_m(\mathbf{s}_i)) & \sin(\theta_m(\mathbf{s}_i)) \\ -\sin(\theta_m(\mathbf{s}_i)) & \cos(\theta_m(\mathbf{s}_i)) \end{bmatrix}. \qquad \text{((xxiii)}$$

For applications where only the bathymetry is needed, $\gamma_D(\mathbf{s}_i, \mathbf{s}_j)$, may be sufficient for use. In hydrographic situations, however, where extra caution is required for navigation safety, an additional "hydrographic uncertainty" variogram is added to increase the total uncertainty for safety (see Calder (2006) for details). This variogram is defined to be

$$2\gamma_H(h) \equiv b_0 + b_1 h \qquad \text{((xxiv)}$$

The choice of the coefficients is arbitrary, but Calder uses $b_0 \equiv 0$ and $b_1 \equiv 6.51 \times 10^{-4}\,\text{m}$ so that the 95% confidence interval of the uncertainty increases by 0.05 m for every meter of horizontal separation in soundings. Thus, for hydrography, the final variogram to use is

$$2\gamma_{total}(\mathbf{s}_i, \mathbf{s}_j) = 2\gamma_D(\mathbf{s}_i, \mathbf{s}_j) + 2\gamma_H(\left\| \mathbf{s}_j - \mathbf{s}_i \right\|) \qquad \text{((xxv)}$$

3. Monte Carlo Estimation of Depth Error Due to Positional Stability Errors

This last part is of use for fusion with soundings data, either with other soundings data or a historical grid, as the position of the soundings contains errors (this part is not applied when fusing historical gridded data sets). As before (c.f. Section I.A), positioning errors result in errors of depth estimates on the interpolation surface as the position of the soundings affect the interpolated solution; the Monte Carlo technique of Jakobbsson et al. (2002) is used to estimate this error. At the beginning of each iteration, the locations of the soundings are perturbed according to a probability density function appropriate for the sounding, often assumed to be Gaussian unless otherwise known (c.f. Fig. 5 in Calder (2006) for a non-Gaussian example). Loess interpolation and kriging, using the variogram from the unperturbed set, are repeated at each iteration. The standard deviation of the solutions at each interpolation point provides $\varepsilon(\mathbf{s})$ in Eqn. (((x).

Acknowledgements

This work is funded through the Office of Naval Research, Naval Research Laboratory base program. The mention of commercial products or the use of company names does not in any way imply endorsement by the U.S. Navy.

References

(2008). Netica, Version 4.0.8. Vancouver, BC, Norsys Software Corporation.
Brandt, S. (1998). Data analysis : statistical and computational methods for scientists and engineers. New York, Springer.
Brandt, S. (1999). Data Analysis: Statistical and Computational Methods for Scientists and Engineers New York, Springer.
Calder, B. (2006). "On the uncertainty of archive hydrographic data sets." IEEE Journal of Oceanic Engineering **31**(2): 249-265.
Cleveland, W. S. (1979). "Robust locally weighted regression and smoothing scatterplots." Journal of the American Statistical Association **74**(368): 829-836.
Cleveland, W. S. and S. J. Devlin (1988). "Locally weighted regression - an approach to regression-analysis by local fitting
" Journal of the American Statistical Association **83**(403): 596-610.
Cressie, N. A. C. (1993). Statistics for Spatial Data. New York, Wiley.
Davis, J. C. (2002). Statistics and Data Analysis in Geology. New York, Wiley.
Elmore, P. A., D. H. Fabre, et al. (2009). Uncertainty estimation of historical bathymetric data from Bayesian networks. OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges.
Givens, G. H. and J. A. Hoeting (2005). Computational Statistics. Hoboken, NJ, Wiley-Interscience.
Goldstein, H. (1980). Classical Mechanics. Reading, MA, Addison Wesley.
Jakobsson, M., B. Calder, et al. (2002). "On the effect of random errors in gridded bathymetric compilations." Journal of Geophysical Research-Solid Earth **107**(B12): Article 2358.
Mahalanobis, P. C. (1936). "On the generalised distance in statistics." Proceedings of the National Institute of Science of India **12**: 49-55.
Marquardt, D. W. (1963). "An algorithm for least-squares estimation of nonlinear parameters." Journal of the Society for Industrial and Applied Mathematics **11**(2): 431-441.

Marsaglia, G. and W. W. Tsang (2000). "The Ziggurat Method for Generating Random Variables." Journal of Statistical Software **5**.
Oksanen, J. and T. Sarjakoski (2005). "Error propagation of DEM-based surface derivatives." Computers & Geosciences **31**(8): 1015-1027.
Plant, N. G., K. T. Holland, et al. (2002). "Analysis of the scale of errors in nearshore bathymetric data." Marine Geology **191**(1-2): 71-86.
Russell, S. J. and P. Norvig (2003). Artificial Intelligence: A Modern Approach. Upper Saddle River, N.J., Prentice Hall/Pearson Education.
Zhou, Q. M. and X. J. Liu (2004). "Error analysis on grid-based slope and aspect algorithms." Photogrammetric Engineering and Remote Sensing **70**(8): 957-962.

10.2.6 Error Growth Model

The following has been extracted from 5th Edition, S-44 (2008) Annex A, section A.5 by Rob Hare and edited for context

TPU is a combination of random and bias based uncertainties. Random and short period uncertainties have to be recognized and evaluated both in horizontal and vertical directions.

The propagated uncertainty may be expressed as a variance (in metres^2) but is more often reported as an *uncertainty* (in metres) derived from variance with the assumption that the uncertainty follows a known distribution. In the latter case, the confidence level (e.g., 95%) and the assumed distribution shall be documented. Horizontal uncertainties are generally expressed as a single value at a 95% level, implying an isotropic (circular) distribution of uncertainty on the horizontal plane.

In the hydrographic survey process it is necessary to model certain long period or constant factors related to the physical environment (e.g. tides, sound speed, vessel dynamic draft, including squat of the survey vessel). Inadequate models may lead to bias type uncertainties in the survey results. These uncertainties shall be evaluated separately from random type uncertainties.

TPU is the resultant of these two main uncertainties. The conservative way of calculating the result is the arithmetic sum, although users should be aware that this may significantly overestimate the total uncertainty. Most practitioners, and the appropriate ISO standard, recommend quadratic summation (i.e., summation of suitably scaled variances).

An example is given in S-44 for fixed (a) and depth-variable (b) uncertainty components:

$$\text{TVU} = \pm\sqrt{a^2 + (b \times d)^2}$$

10.2.7 Split-Sample Approach

Contributed by Chris Amante and Matt Love, NOAA National Geophysical Data Center (NGDC)

There are a number of techniques used to quantify the uncertainty of interpolated elevations, e.g., split-sample, cross-validation, jack-knifing, and boot-strapping. Using a split-sample approach, a percentage of the data is omitted, an interpolation method is applied, and the differences between the interpolated elevations and the original omitted elevations are calculated. This method is often used to assess the stability of various interpolation methods by omitting increasingly greater percentages of the original data and analyzing changes in the uncertainty.

A program for Linux users written in Guile Scheme (http://www.gnu.org/s/guile/) has been developed to quantify the uncertainty of interpolation methods using a split-sample approach. The program, named ss_unc.sch, utilizes the Generic Mapping Tools (GMT; http://www.soest.hawaii.edu/gmt/), and ancillary programs written in the C programming language by Matthew Love of the National Oceanic and Atmospheric Administration (NOAA) National Geophysical Data Center (NGDC; see glossary).

The program omits a percentage of the xyz points, applies an interpolation method, and calculates the differences between the interpolated values and the omitted elevations. In order to quantify the uncertainty of the interpolation method at every data point, the program repeats this process and aggregates the differences between the original xyz file and the interpolated elevations. The program produces several useful products to be used in assessing the uncertainty of various interpolation methods. The products include a histogram (see Figure 10.2.7.1) of the differences with statistical measurements such as the minimum, maximum, mean, root mean squared error (RMSE), and standard deviation. In addition, the program produces a binary .grd file of the differences between the original and interpolated elevations.

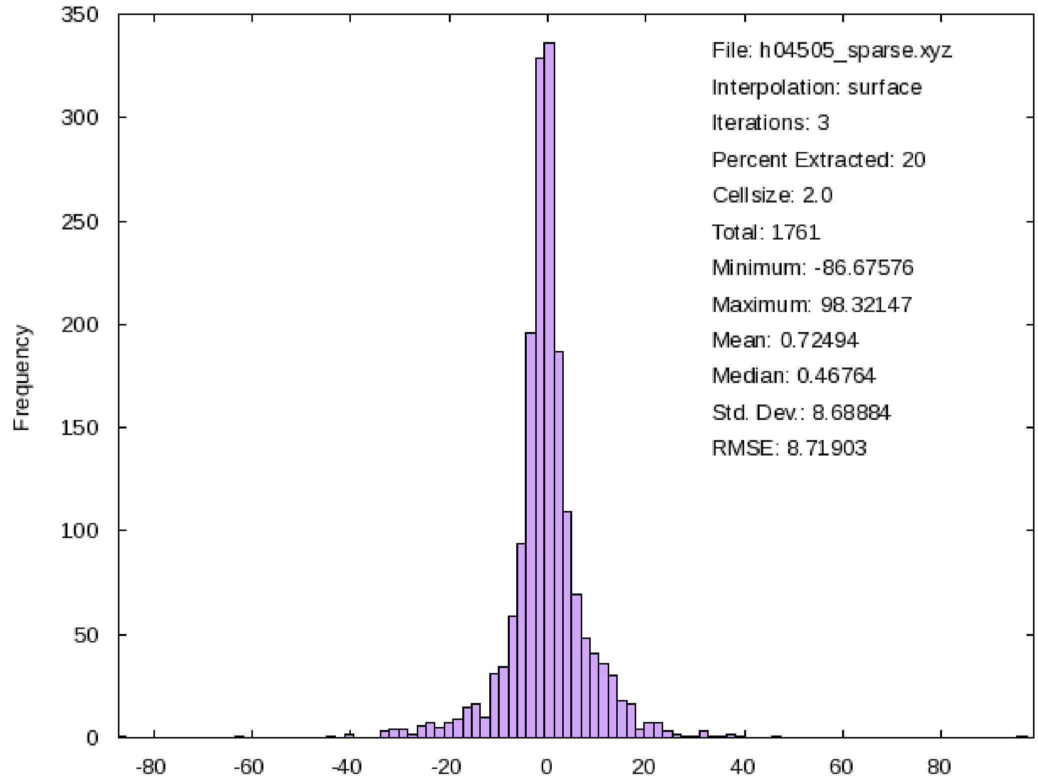


Figure 10.10 Example histogram output from Split-Sample program SS_UNC.

The program has multiple user-defined parameters. The user defines the input xyz file to be used to evaluate multiple gridding methods, the gridding methods to evaluate, the percentage of points to be omitted, the grid cell size, the input xyz file delimiter, and the number of iterations to repeat the process (see example below). Currently the program can evaluate GMT 'surface,' 'nearneighbor,' and 'triangulate' gridding algorithms. Additional gridding algorithms, such as MB-System 'mbgrid,' will be included in the program in the near future.

ss_unc.scm [infile] [-e arg] [-percent arg] [-grd_cells arg] [-delimiter arg] [-dotimes arg] [-engines? arg] [-help]

infile	The input xyz file
-e	Specify which gridding engine(s) to use. If multiple engines are desired, separate them with commas.
-percent	Percentage of xyz data to extract, default is 20.
-grd_cells	The grid cell size in arc-seconds.
-delimiter	The input xyz delimiter
-dotimes	The number of iterations
-engines?	Check available engines
-help	Display the help

Split-sample uncertainty program example:

~] ss_unc.scm h04505_sparse.xyz -e surface, nearneighbor, triangulate -percent 25 -grd_cells 100 -delimiter "," -dotimes 10 > ss_unc_example.sh

~] chmod +x ss_unc_example.sh

~] ./ss_unc_example.sh

Another option currently in progress is to evaluate the stability of the interpolation method by omitting an increasingly greater percentage of original data. For example, omit 20% of the original data and increase the percentage of omitted data at 10% intervals until 80% of the original data has been omitted. This option will produce a line graph depicting the change in RMSE as a function of the percentage of points omitted for various interpolation methods.

SS_UNC

Compare interpolation gridding methods to help determine uncertainty.

Matthew Love

1 Copying

SS UNC is copyright (C) 2011 CIRES and is released under the terms of the GNU General Public License. See the included file 'COPYING' for the full text of the license (or see section "Copying" in The GNU Emacs Manual).

> This is free software – you are welcome to redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.
>
> SS UNC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

2 Introduction

The Split Sample UNCertainty program (SS UNC) is a Guile scheme command-line program designed to run on the GNU/Linux Operating System. SS UNC is highly configurable, allowing the user to add gridding engines, change and add command-line options, and even change the way SS UNC runs.

3 Installing SS UNC

3.1 Dependencies

For SS UNC to work, there are a number of required dependencies:

- GNU/Linux
- Guile
- GMT
- rpslit
- ncquery

3.2 Autotools

Installation through autotools is fairly straight forward.

1. Unpack the package

   ~] tar xfvz ss_unc-0.1.tar.gz

2. Enter the package root directory

   ~] cd ss_unc-0.1

3. Configure and Make the package

   The '--prefix' switch to ./configure specifies where to install files.

   ~/ss_unc-0.1] ./configure --prefix=~/ss_unc && make

4. Install the built package

   ~/ss_unc-0.1] make install

3.3 Manual Installation

1. Unpack the package

   ~] tar xfvz ss_unc-0.1.tar.gz

2. Enter the package root directory

   ~] cd ss_unc-0.1

3. Enter the package 'src directory

   ~] cd src

4. Copy ss_unc.scm to a directory locatable with PATH

   ~] cp ss_unc.scm ~/bin

4 Using SS UNC

SS UNC is a command-line program designed to be run in a shell program such as bash.

4.1 Basic Usage

For general usage help from the command-line, the '-help' switch can be used. The usage display will display the default command-line switches as well as any additional switches defined by the user, see Section 5.2 [Adding Command Line Options], page 6. The '-e' switch specifies the gridding engine to use in gridding comparisons, see Section 5.1 [Adding Engines], page 6.

```
~] ss_unc.scm -help
ss_unc.scm version 0.2.0
Usage: ss_unc.scm [infile] [-e arg] [-percent arg] [-grd_cells arg]
                  [-delimiter arg] [-dotimes arg] [-engines? arg] [-help]

infile            The input xyz file.
-delimiter        The input xyz record delimiter, default is space (‘‘ ‘‘)
-e                Specify which gridding engine(s) to use,
                  if multiple engines are desired, separate them with commas.
-percent          Percentage of xyz data to extract, default
                  is 20, separate values with a comma.
-grd_cells        The grid cell size -dotimes The number of iterations -engines?
                  Check available engines, optionally specify
                  which engine to get more information about..
-help             Display the help
```

infile

The infile is the input xyz file to perform the gridding algorithms on.

-e

The -e switch specifies which gridding engine(s) to use. To use multiple engines, separate the engine names with a comma, this will produce histograms and stat files for each of the engines mentioned.

-percent

The -percent switch specifies the percentage of random points to extract from infile. If the argument is a list of comma separated values, ss unc.scm will run for each of the specified percentages, and append the statistics for each into a separate file.

-grd cells

The -grd_cells switch specifies the cell-size to use for gridding operations.

-dotimes

The -dotimes switch allows the user to specify how many iterations of gridding and comparing will be performed.

-engines?

The -engines? switch without arguments will display a list of the available gridding engines, optionally specify an engine as an argument to get more details about that engine.

-help

This switch will display the help text and exit.

When ss unc.scm is run, it will immediately output a number of commands to stdin. If these commands are redirected to a file, e.g. > ss_unc_test.sh, then that file can be used to generate the output files. The generated script can be edited to fine-tune some of the commands if needed and can be saved for future use.

4.2 Default Engines

SS UNC comes with a number of gridding engines for general use. To check which engines are available on the running version of SS UNC, use the '-engines? switch, which will display all the engine keys which would be used with the '-e option, including those added by the user, see Section 5.1 [Adding Engines], page 6.

Each of the search engines has a description, which can be displayed by specifying which engine to display more information about after the '-engines? switch:

~] ss_unc.scm -engines?

surface

nearneighbor

triangulate

~] ss_unc.scm -engines? surface

surface

GMT surface (SPLINE)

4.3 Output

SS UNC will generate a number of output files. The type and number of output files will depend on the given command-line arguments.

- Histogram(s)
- Difference Grid(s)
- Statistics file(s)

5 Extending SS UNC

SS UNC allows for complete extendibility. By use of hooks and definitions, one could rewrite SS UNC to ones own desires.

5.1 Adding Engines

SS UNC allows for the adding of new gridding engines. This is accomplished through the configuration file (~/.ss unc). The scheme function ’ci-add-engine is used to add a new gridding engine into SS UNC:

```
(si-add-engine ’(engine-name .
                 (’engine-function ‘‘engine-description’’)))
```

where engine-function is a function which should accept 2 arguments (input-xyz-file and gridding-cell-size) and engine-description is a string describing the engine. Adding an engine in this manner will add the engine to the output of the -engines? command-line switch, seamlessly integrating the added engines into the SS UNC interface.

5.2 Adding Command Line Options

SS UNC allows for the adding of new command-line options. There are two steps involved in adding command-line options, adding the option to the help menu and adding a handling function to the command-line hook.

```
(define switch-var #f)

   (sc-add-option ’(-switch . (takes-options?(#t or #f)
   "Description"))) (add-hook! ci-command-line-hook
   (lambda (a b)
         (if (equal? (car b) "-show_url")
 (begin
    (set! show-url #t)
    (parse-options a (cdr b))))))
```

5.3 Hooks

Hooks are a useful tool in extending scheme code. SS UNC takes advantage of hooks to allow the user the ability to add command-line options (including in the -help output).

SS UNC provides one hook, listed and described below:

ci-command-line-hook

The ’sc-command-line-hook is the hook that allows adding to the default command-line options. The hook is run before the general command-line parsing begins, allowing the user to have control over the command-line environment. This hook is passed 2 options, the first is the command that ran the program (e.g. /usr/bin/sicl) and the second is the remaining command-line options. The hook will be run for each command-line option, so the user doesn’t have to write any loops in the configuration files to take advantage of adding command-line options in this manner.

6 Examples

Running ss unc.scm

Run ss unc.scm on the comma delimited ascii xyz file test.xyz, randomly extracting 25 percent of the data points, gridding using surface and triangulate at 10 arc-seconds.

```
#run the command
~]   ss_unc.scm test.xyz -percent 25 -e surface,triangulate \

-grd_cells 10 -delimiter ‘‘,’’ > test.sh

#Allow the output script to be executable
~]   chmod +x test.sh

#Run the output script
~] ./test.sh
```

Re-Define the ’surface engine function

Redefine the ’surface engine function in the SS UNC configuration file, adding a tension parameter.

```
(define (run-surface in cellsize)
    (display (string-append
                    "surface "
                    in
                    " -I"
                    cellsize
                    "c   -T.4 -G"
                    (basename in ".xyz") "_surface.grd"
                    " $(minmax " in " -I" cellsize "c)"))
    (newline))
```

SS_UNC.SCM

SS_UNC


```
#!/usr/bin/guile \
−e main −s
!#

;;−−

(use−modules (ice−9 rdelim))

(define percent "20")
(define inxyz '())
(define grdcellsize "1")
(define do−times "2")
(define outps "out.ps")
(define delim " ")
(define version−wanted? #f)
(define help−wanted? #f)

;;−−−−−−−−−−;;
;; Engines ;;
;;−−−−−−−−−−;;
;;−−−
;; Define some hooks.

(define ci−command−line−hook (make−hook 2))

;; The User Config File, this can be set in the .guile config file.

                 (if (not (defined? 'ci−config))
                         (define ci−config
                         (cond ((file−exists? (string−append (getenv "HOME") "/.ss_unc.conf"))
                        (string−append (getenv "HOME") "/.ss_unc.conf"))
                        ((file−exists? (string−append (getenv "HOME") "/.ss_unc"))
                        (string−append (getenv "HOME") "/.ss_unc")))))

;; The list of default engines. Formatted
;; thusly: '(name . ('function "description")
;; The 'function should accept 2 arguments,
;; the input xyz file and the gridding cellsize

(define ci−engine−alist

    '((surface . ('run−surface "GMT surface (SPLINE) \n"))
        (triangulate . ('run−triangulate "GMT triangulate (Triangulation) \n"))
        (nearneighbor . ('run−nn "GMT nearneighbor (NEAREST NEIGHBOR) \n"))))

(define (ci−add−engine specs)

    "Add a gridding engine to the 'ci−engine−alist in the format
'(name . 'function \"description\")"

    (append! ci−engine−alist (list specs)))

(define (ci−display−engines engine−alist)
```

```
  "Display the available engines in the given 'engine-alist."

  (if (pair? engine-alist)

      (begin
        (display (caar engine-alist))
        (newline)
        (ci-display-engines (cdr engine-alist)))))

(define (ci-describe-engine ci-eng engine-alist)

  "Describe 'ci-eng using the given 'engine-alist"

  (display ci-eng)
  (newline)
  (display "--------")
  (newline)
  (if (and (assq (string->symbol ci-eng) engine-alist)

           (not (null? (cddr (assq (string->symbol ci-eng) engine-alist)))))
      (display (caddr (assq (string->symbol ci-eng) engine-alist)))
      (display "Sorry, this engine has no description available."))

  (newline))

;;---

;;---Engine functions:

(define (run-surface in cellsize)
  (display (string-append

           "surface "

           in

           " -I"

           cellsize

           "c -G"
           (basename in ".xyz") "_surface.grd"
           " $(minmax " in " -I" cellsize "c)"))

  (newline))

(define (run-nn in cellsize)
  (display (string-append

           "nearneighbor "

           in

           " -I"
```

```
            cellsize

            "c -S20c -N8 -V -G"
            (basename in ".xyz") "_nearneighbor.grd"
            " $(minmax " in " -I" cellsize "c)"))

   (newline))

(define (run-triangulate in cellsize)
   (display (string-append

            "triangulate "

            in

            " -I"

            cellsize

            "c -G"
            (basename in ".xyz") "_triangulate.grd"
            " $(minmax " in " -I" cellsize "c) > tmp.idk"))

   (newline))

;;---

;;---PROC:

(define (run-rsplit in percentage)
   (display (string-append

             "rsplit --percent "

             percentage
             " -r " in " > rand_"
             (basename in)

             " 2>base_"

             (basename in)))
   (newline))

(define (run-ncquery ingrd inxyz delim d_format outxyz)
   (display (string-append

             "ncquery -r "

             d_format

             " -g "
```

```
                    ingrd
                    " −i "
                    inxyz
                    " −d "
                    delim
                    " | grep −v \"nan\" >> "
                    outxyz))
    (newline))

(define (run−xyz2grd inxyz outgrd cellsize)
    (display (string−append
                    "awk −F"
                    delim
                    " '{print $1,$2,$5}' | "
                    "xyz2grd "
                    inxyz
                    " −V −I"
                    cellsize
                    "c −G"
            outgrd
            " $(minmax " inxyz " −I" cellsize "c)"))
             (newline))

;;−−−STATS:

    (define (run−pshistogram inxyz outps rperc ptimes engine)
        (display
        (string−append
        "pshistogram "
        inxyz
```

```
" −V −Gblue −W$(minmax −C "
 inxyz " | awk '{print ($10−$9)*.10}') −L1 −Z0 −T4
−B:\"Value\":/:\"Frequency\"::.:WSne −U/−.2
5i/−0.75i/\"ss_unc.scm\" −K > "
 outps " 2>" outps ".info"))
 (newline))

(define (run−pstext inxyz outps
    rperc ptimes engine)
     (display
    (string−append

   "pstext −R0/3/0/5 −JX3i −O −N −V << EOF >> "

    outps
   "\n.25 11.5 10 0 0 LT Iterations: " ptimes
   "\n.25 11 10 0 0 LT Percent: " rperc
   "\n.25 10.5 10 0 0 LT Cell−Size: " grdcellsize
   "\n1.5 11.5 10 0 0 LT Total: $(wc −l " inxyz " | awk '{print $1}')"
   "\n1.5 11 10 0 0 LT Min: $(grep min/max " outps ".info | awk '{print $6}')"
   "\n1.5 10.5 10 0 0 LT Max: $(grep min/max " outps ".info | awk '{print $7}')"
   "\n2.75 11.5 10 0 0 LT Mean: $(tail −1 "
    (basename inxyz ".xyzgd") ".stat | awk '{print $4}')"
   "\n2.75 11 10 0 0 LT RMSE: $(tail −1 "
    (basename inxyz ".xyzgd") ".stat | awk '{print $5}')"
   "\n4 11.5 10 0 0 LT File: " inxyz
   "\n4 11 10 0 0 LT Interpolation: " engine
   " \nEOF"))

  (newline))

(define (set−gmt−defaults)

  (display "gmtset COLOR_NAN 255/255/255 DOTS_PR_INCH 300 ANNOT_FONT_PRIMARY Times−Roman \
ANNOT_FONT_SIZE_PRIMARY 8 ANNOT_FONT_SIZE_SECONDARY 8 HEADER_FONT Times−Roman \
LABEL_FONT Times−Roman LABEL_FONT_SIZE 8 HEADER_FONT_SIZE 10 PAPER_MEDIA letter \
UNIX_TIME_POS 0i/−1i PS_COLOR CMYK Y_AXIS_TYPE ver_text PLOT_DEGREE_FORMAT ddd:mm:ss \
D_FORMAT %.1f COLOR_BACKGROUND 0/0/0 COLOR_FOREGROUND 255/255/255 COLOR_NAN 255/255/255")

  (newline))

(define (run−stats inxyz outxyz rperc ptimes grdcellsize)

  (display (string−append
              "awk −F" delim " '{sum+=$5} {sum2+=($5*$5)} END { print \""
              rperc "\",\"" ptimes "\",\"" grdcellsize "\",sum/NR,sqrt(sum2/NR)}' "
              inxyz " >> " outxyz))

  (newline))

     (define (run−ps2raster psfile outf)
        (display (string−append
```

```
               "ps2raster −T" outf " " psfile))
               (newline))

;;−−−

;;−−−−−−−−−−−−−−−−;;
;; Run−Commands ;;
;;−−−−−−−−−−−−−−−−;;
;;−−−Run through the commands

        (define (run−commands
            run−times engines p)
             (let ((engine−list
            (string−split engines #\,)))
             (if (> run−times 0)
            (begin
            (let ((base−xyz
            (string−append "base_"
            (basename inxyz)))

                        (rand−xyz (string−append "rand_" (basename inxyz))))
                  (run−rsplit inxyz p)
                  (map (lambda (x)

                              (primitive−eval
                                (list (cadr (cadr (assq
                                (string−>symbol x)
                                ci−engine−alist)))
                                base−xyz grdcellsize)))
                                engine−list)
                                (map (lambda (x)

                              (run−ncquery
                                (string−append (basename base−xyz ".xyz") "_" x ".grd")
                                rand−xyz
                                delim

                                "xyzgd"
                                (string−append "rand_" p "_"

                                                                    (basename
                                      inxyz ".xyz") "_" x ".xyzgd")))
                                      engine−list)
                                      (run−commands (− run−times 1)
                                      engines p)))
                                      (begin
                                      (map (lambda (x)
                                      (run−xyz2grd
                                      (string−append "rand_" p "_"
                                      (basename inxyz ".xyz") "_" x
                                      ".xyzgd")
                                      (string−append "rand_" p "_"
                                      (basename inxyz ".xyz") "_" x
                                      ".xyzgd.grd")
                                      grdcellsize))
                                      engine−list)
                                      (map (lambda (x)
                                      (run−stats (string−append "rand_" p
                                      "_"
                                      (basename inxyz ".xyz") "_" x
                                      ".xyzgd")
                                      (string−append "rand_"
                                      (basename inxyz ".xyz") "_" x ".stat")
```

```
                    p do-times grdcellsize)
                    (run-stats (string-append "rand_" p
                    "_"
                    (basename inxyz ".xyz") "_" x
                    ".xyzgd")
                    (string-append "rand_" p "_"
                    (basename inxyz ".xyz") "_" x ".stat")
                    p do-times grdcellsize))

            engine-list)
        (set-gmt-defaults)
        (map (lambda (x)

              (run-pshistogram
               (string-append "rand_" p "_" (basename inxyz ".xyz") "_" x ".xyzgd")
               (string-append "rand_" p "_" (basename inxyz ".xyz") "_" x ".ps")
               p do-times x))

            engine-list)
        (map (lambda (x)

              (run-pstext
               (string-append "rand_" p "_" (basename inxyz ".xyz") "_" x ".xyzgd")
               (string-append "rand_" p "_" (basename inxyz ".xyz") "_" x ".ps")
               p do-times x))

            engine-list)
        (map (lambda (x)

              (run-ps2raster
               (string-append "rand_" p "_" (basename inxyz ".xyz") "_" x ".ps")
               "G"))

            engine-list)))))

;;---

;;-----------------------;;
;; Command-line options ;;
;;-----------------------;;

;;---Options
;; '(switch . (args? "desc"))

(define ci-option-alist '((infile . (#f "The input xyz file."))
                          (-e . (#t "Specify which gridding engine(s) to use, \

if multiple engines are desired, separeate them with commas."))
                          (-percent . (#t "Petcentage of xyz data to extract, default is 20"))
                          (-grd_cells . (#t "The grid cell size"))
                          (-delimiter . (#t "The input xyz delimiter"))
                          (-dotimes . (#t "The number of iterations"))
                          (-engines? . (#t "Check available engines, optionally specify \

which engine to get more information about.."))
                          (-help . (#f "Display the help"))))

(define (ci-add-option specs)
  (append! ci-option-alist (list specs)))

    (define (ci-display-options
      option-list)
```

```
        (if (pair? option−list)
        (begin

          (if (cadar option−list)
                (display (string−append " [" (symbol−>string (caar option−list)) " arg]"))
                (display (string−append " [" (symbol−>string (caar option−list)) "]")))

          (ci−display−options (cdr option−list)))
        (display "\n\n")))

          (define (ci−describe−options option−list)
              (if (pair? option−list)
              (begin
             (let ((cname (symbol−>string (caar option−list)))
             (cdesc (caddr (car option−list))))
             (if (> 20 (string−length cname))

                 (display (string−append

                                  (string−append
                                    cname
                                    (make−string (− 20 (string−length cname)) #\ ))

                                 "\t" cdesc "\n"))))
         (ci−describe−options (cdr option−list)))
          (display "\n")))

(define (display−help command−name option−list)
    (if (pair? option−list)

        (begin
          (display (basename command−name))
          (display " version 0.2.1")
          (newline)
          (display "Usage: ")
          (display (basename command−name))
          (ci−display−options option−list)
          (ci−describe−options option−list))))

;; Read the given port ('p) and put it into the string 'pstring.


(define (read−port p pstring)


    (if (not (eof−object? (peek−char p)))
        (read−port p (string−append pstring (read−line p)))
        pstring))

;; Parse the command−line


         (define (parse−options cca ccl)
              (if (pair? ccl)
              (begin
             (if (not (hook−empty? ci−command−line−hook))
             (run−hook ci−command−line−hook cca ccl))
             (cond


            ;; Help? ;;


            ((or (equal? (car ccl) "−h")
                  (equal? (car ccl) "−help")
                  (equal? (car ccl) "−?"))

              (display−help cca ci−option−alist)
```

```
          (exit 1))

        ((equal? (car ccl) "-percent")

          (begin
             (set! percent (cadr ccl))
             (parse-options cca (cddr ccl))))

        ;; Engines ;;

        ((equal? (car ccl) "-engines?")

          (if (not (null? (cdr ccl)))
                (ci-describe-engine (cadr ccl) ci-engine-alist)
                (ci-display-engines ci-engine-alist))

          (exit 1))

        ;; Gridding Engine ;;

        ((equal? (car ccl) "-e")

          (begin
             (set! ci-engine (cadr ccl))
             (parse-options cca (cddr ccl))))

        ((equal? (car ccl) "-delimiter")

          (begin
             (set! delim (cadr ccl))
             (parse-options cca (cddr ccl))))

        ((equal? (car ccl) "-grd_cells")
          (begin

             (set! grdcellsize (cadr ccl))
             (parse-options cca (cddr ccl))))

        ((equal? (car ccl) "-dotimes")

          (begin
             (set! do-times (cadr ccl))
             (parse-options cca (cddr ccl))))

        ;; Input xyz file ;;

        ((null? inxyz)
          (set! inxyz (car ccl))
          (parse-options cca (cdr ccl)))

        (else
          (parse-options cca (cdr ccl))))))

;; Load the User Config File
;(display ci-config)

     (if (file-exists? ci-config)
          (begin
          (load ci-config)))
```

```
(if (not (defined? 'ci−engine))
        (define ci−engine "surface"))

(define (main args)

    ;; Parse the command−line

     (parse−options (car (command−line)) (cdr (command−line)))

    ;; If 'inxyz was not given, get it from 'current−input−port

     (if (null? inxyz)
            (set! inxyz (read−port (current−input−port) "")))

    ;(display inxyz)

                (let ((percent−list (string−split percent #\,)))
                    (map (lambda (p)
                   (run−commands (string−>number do−times) ci−engine p))
                   percent−list)))

;;−−END
```

RSPLIT

Randomly extract lines from a text file.
Manual

Matthew Love

1 Copying

2 Introduction

Randomly extract a percentage of lines from a text file. The randomly extracted line gets printed to stdout, optionally, the remaining lines can be printed to stderr, allowing for the splitting of text files.

3 Installing rsplit

Installation through autotools is fairly straight forward.

1. Unpack the package
   ~] tar xfvz rsplit-0.1.3.tar.gz

2. Enter the package root directory
   ~] cd rsplit-0.1.3

3. Configure and Make the package
   The '--prefix' switch to ./configure specifies where to install files.
   ~/rsplit-0.1.3] ./configure --prefix=~/rsplit && make

4. Install the built package
   ~/rsplit] make install

4 Using rsplit

RSPLIT is a command-line program designed to be run in a shell program such as bash. For general usage help from the command-line, the '--help' switch can be used.

~] rsplit --help
Usage: rpslit [OPTIONS] [infile]
Randomly extract a percentage of lines from a text file.
Options:
-p, --percent The percent of lines to randomly extract
from the input.
-l, --line-num Include line numbers in output.
-r, --remaining Send the lines that aren't extracted to stderr.
--version Print version information and exit.
Example: rsplit test.xyz -p 20 -lr > test_rand20.xyz 2>test_base.xyz

The '-p, --percent' switch specifies the percentage of randomly selected points to extract from the given text file.

The '-l, --line-num' switch prints the line number along with the original line from the input text file.

The '-r, --remaining' switch prints the remaining lines after random extraction to stderr.

5 Examples

Extract 20 percent of the lines in the input text file and print the results to stdout ~] rsplit -p 20 input.txt
Extract 15 percent of the lines in the input text file and write the results to a file, include the line numbers
~] rsplit -p 20 --line-num input.txt > rand20_input.txt
Extract 40 percent of the lines in the input text file, write the results to a file and send the remaining lines to another file. ~] rsplit test.xyz -p 40 -r > test_rand40.xyz 2>test_base60.xyz

10.2.8 Spectral analysis

10.2.9 Slope

10.2.10 Kriging discussion and cautionary note

10.3 Sharing uncertainty results

TEXT IN PREPARATION

10.3.1 ASCII files

TEXT IN PREPARATION

10.3.2 netCDF files

TEXT IN PREPARATION

10.3.3 "BAG" files

Contributed by Rob Hare, Canadian Hydrographic Service, Canada

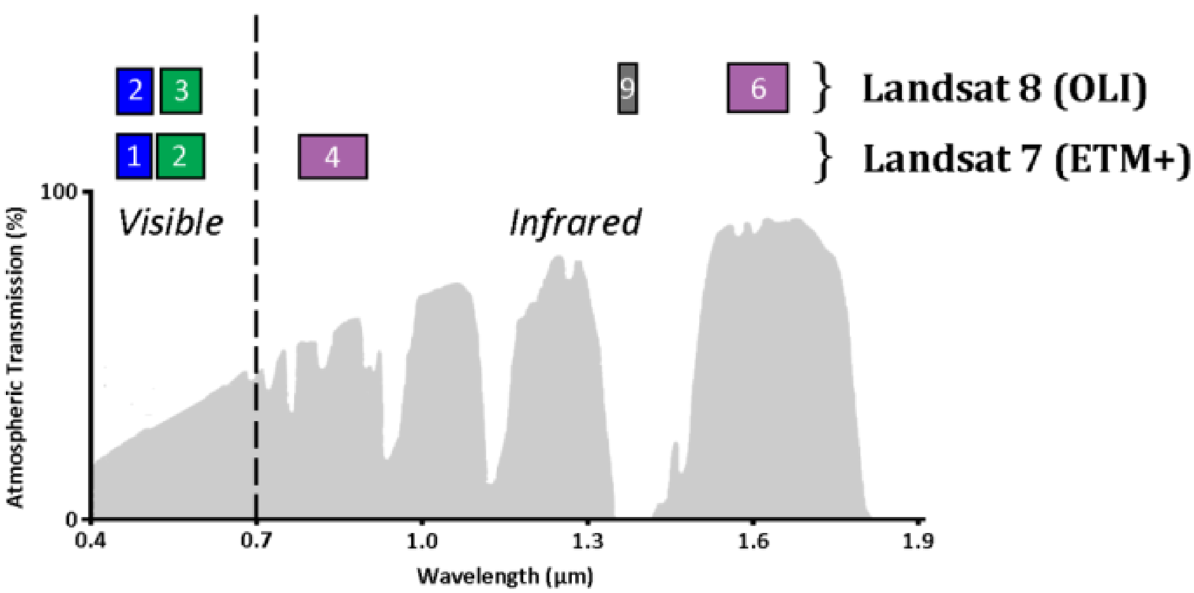
Bathymetric Attributed Grid (BAG) is a non-proprietary file format for storing and exchanging bathymetric data developed by the Open Navigation Surface Working Group. BAG files are gridded, multi-dimensional bathymetric data files that contain position and depth grid data, as well as position and uncertainty grid data, and the metadata specific to that BAG file. Information about BAG files may be obtained from the Open Navigation Surface website:

http://www.opennavsurf.org/whitepapers.html

Chapter 11.0 LANDSAT 8 Satellite-Derived Bathymetry

Contributed by S. Pe'eri, B. Madore and L. Alexander, Center for Coastal and Ocean Mapping, USA, C. Parrish and A. Armstrong, National Oceanic and Atmospheric Administration, USA, C. Azuike, Nigerian Navy Hydrographic Office Lagos, Nigeria, and Eunice N. Tetteh, Ghana National Oceanographic Data Centre,Ghana

Satellite derived bathymetry is a useful reconnaissance tool that can be used to map near-shore bathymetry, characterize a coastal area and to monitor seafloor changes that may have occurred since the last hydrographic survey was conducted. Although there are several commercial multispectral satellite platforms (e.g., Ikonos and WorldView) that can can be used for satellite derived bathymetry, Landsat satellite imagery provides a free and publically available resource. Satellite imagery in the U.S. Geological Survey data archives include also the two most recent Landsat missions were collected using Enhanced Thematic Mapper Plus (ETM+) in Landsat 7 (oparated successfully from April, 1999 until June, 2003) and operational land imager (OLI) in Landsat 8 that is operational since mid-2013. Although the imagery from Landsat 7 and Landsat 8 have the same swath-width of 185 km and an image resolution of 30 m. The number of bands and their spectral range is different in the imagery between the two satellites. The main difference is that Landsat 8 imagery contains an additional band in (Band 9) in the infrared (1.36 - 1.38 µm) that can map cirrus ice clouds. Data from this band can be used to correct some of the atmospheric contribution from the derived bathymetry. The satellite-derived procedure described below provides steps for processing Landsat imagery from both satellites.



Bands of Landsat 7 and 8 imagery used in the SDB procedure.

The key steps in the satellite-derived procedure include:

1. **Pre-processing–** Satellite imagery is downloaded based on the geographic location and environmental conditions (e.g., cloud coverage and sun glint) had to be used.

2. **Water separation–** Dry land and most of the clouds are removed.

3. **Spatial filtering–** ‘Speckle noise’ in the Landsat imagery is removed using spatial filtering.

4. **Glint/cloud correction–** The Hedley et al. (2005) algorithm is used to correct radiometric contributions from sun glint and low clouds.

5. **Applying the bathymetry algorithm–** The bathymetry is calculated using the Stumpf et al. (2003) algorithm on the blue and green bands.

6. **Identifying the extinction depth–** The optic depth limit for inferring bathymetry (also known as, the extinction depth) is calculated.

7. **Vertical referencing–** A statistical analysis between the algorithm values to the chart soundings references the Digital Elevation Model (DEM) to the chart datum.

For more details on satellite-derived bathymetry and hydrographic applications, please refer to the following references:

Hedley, J., A. Harborne and J. Mumby, 2005. Simple and robust removal of sun glint for mapping shallow-water benthos, *International Journal of Remote Sensing*, 26, 2107-2112.

Pe’eri, S., C.Parrish, C.Azuike, L. Alexander and A. Armstrong, 2014. Satellite Remote Sensing as Reconnaissance Tool for Assessing Nautical Chart Adequacy and Completeness, *Marine Geodesy* (accepted).

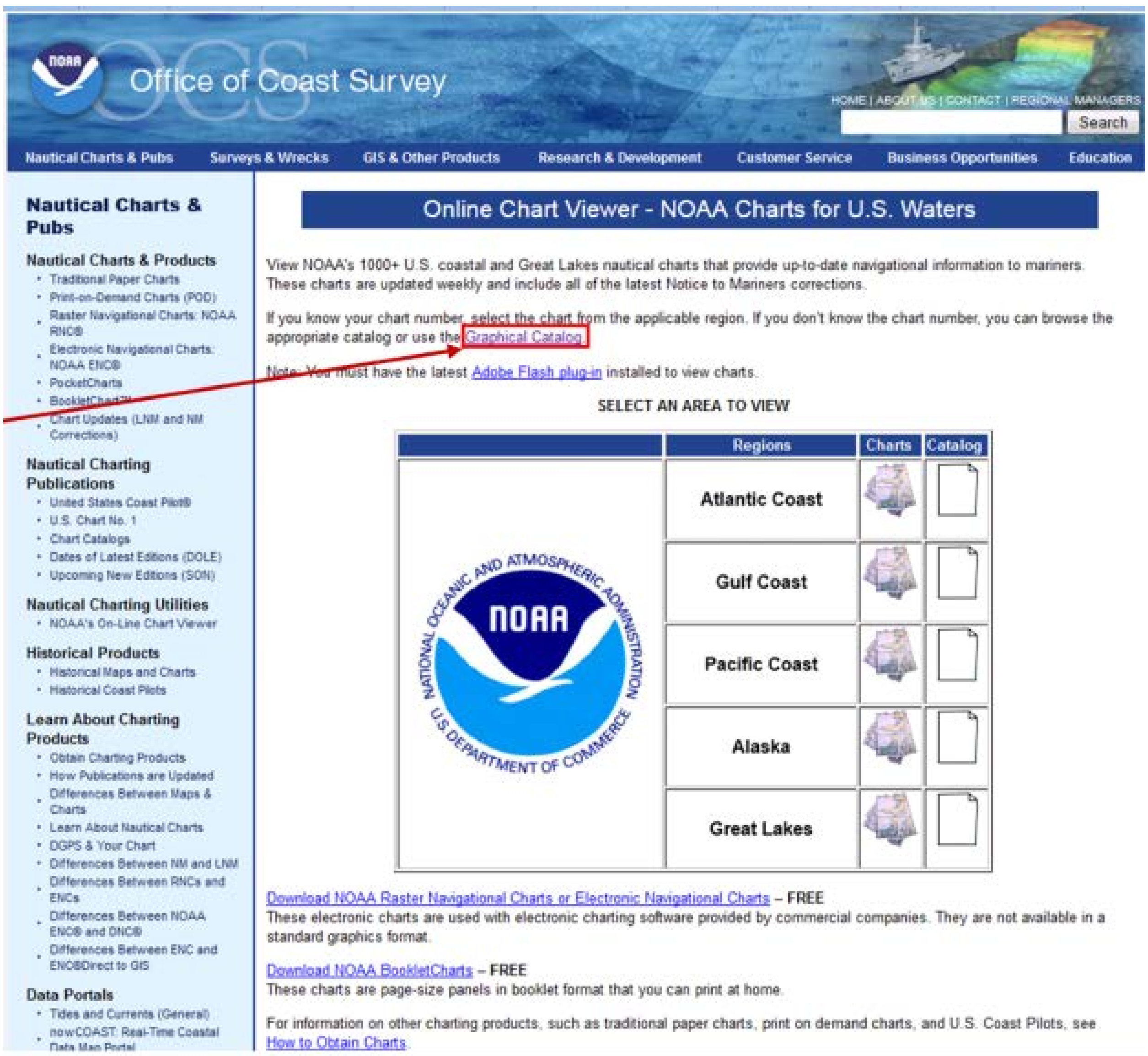
Stumpf, R., K.Holderied and M.Sinclair, 2003, Determination of water depth with high-resolution satellite imagery over variable bottom types, *Limnology and Oceanography*, **48,** 547-556.

11.1 Downloading the datasets

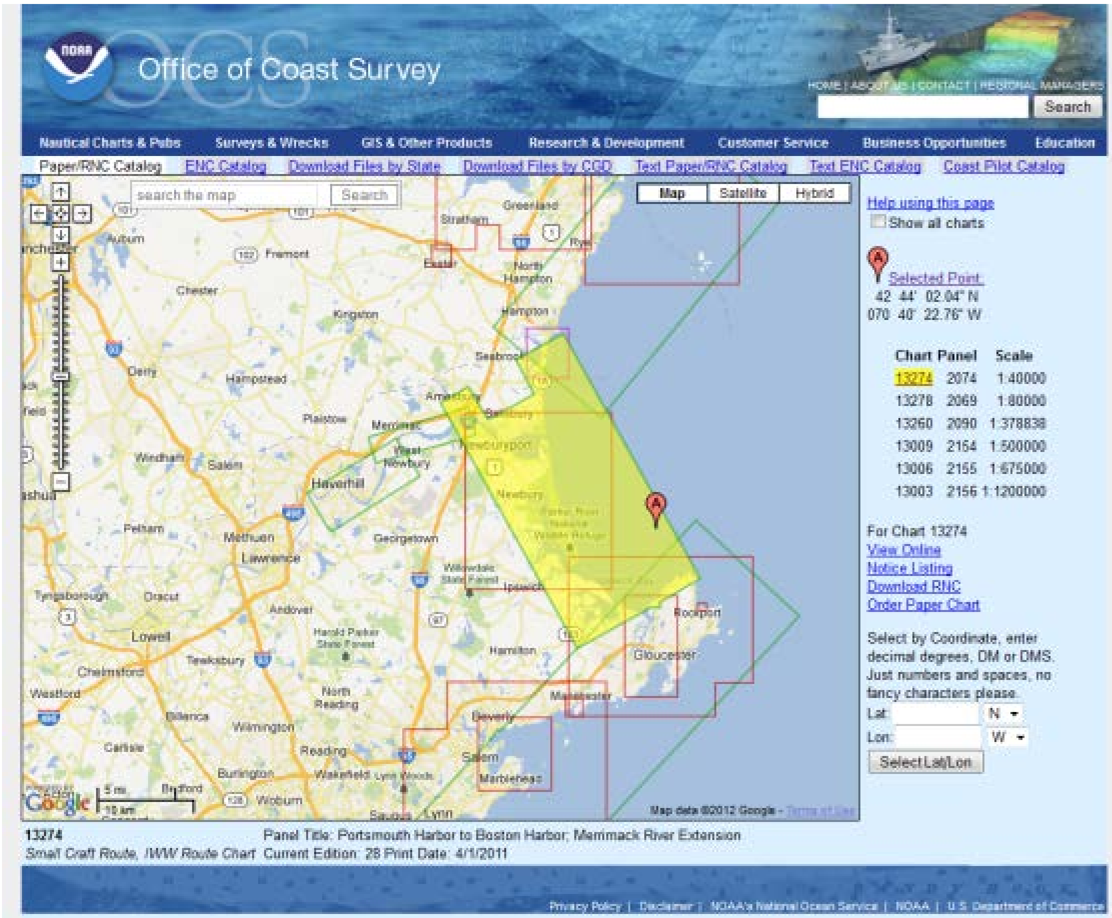
<u>Note</u>: Section 11.1 is for users that are investigating over the U.S. territories (NOAA provides their charts as referenced rasters). In other places around the world, the chart might require scanning and referencing the user should refer to Appendix A.

11.1.1 Downloading a NOAA chart

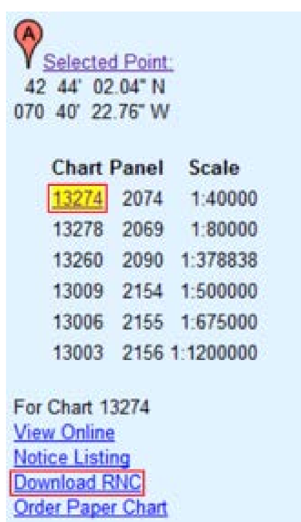
Go to http://www.nauticalcharts.noaa.gov/mcd/NOAAChartViewer.html and click on the ***Graphical Catalog*** link.



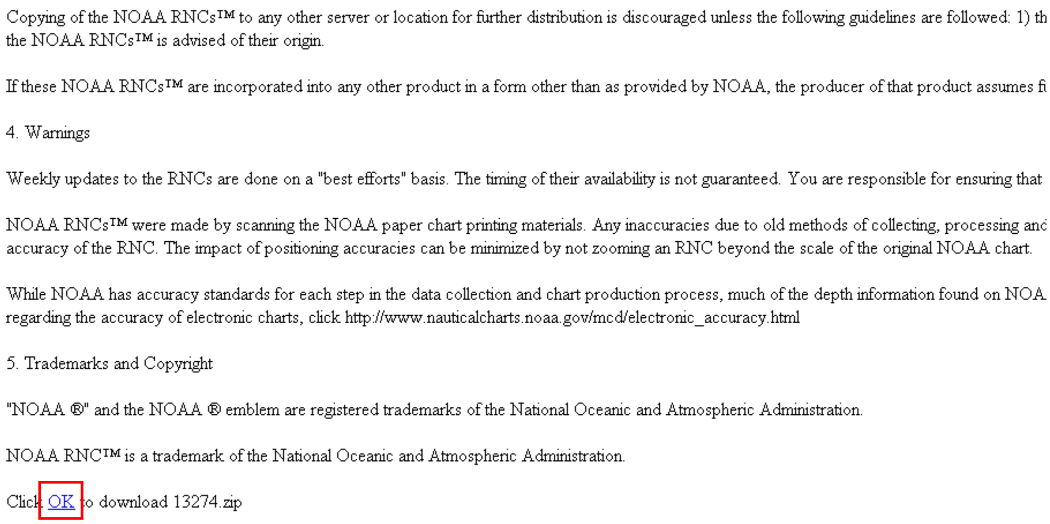
Zoom into the region being examined and click on region where the chart is desired.



On the right side of the screen, make sure that the chart you are interested is selected and click on ***Download RNC***.



A **User Agreement** window will open. Scroll to the bottom of the window and select **OK** to download chart.



Copying of the NOAA RNCs™ to any other server or location for further distribution is discouraged unless the following guidelines are followed: 1) th
the NOAA RNCs™ is advised of their origin.

If these NOAA RNCs™ are incorporated into any other product in a form other than as provided by NOAA, the producer of that product assumes fi

4. Warnings

Weekly updates to the RNCs are done on a "best efforts" basis. The timing of their availability is not guaranteed. You are responsible for ensuring that

NOAA RNCs™ were made by scanning the NOAA paper chart printing materials. Any inaccuracies due to old methods of collecting, processing and
accuracy of the RNC. The impact of positioning accuracies can be minimized by not zooming an RNC beyond the scale of the original NOAA chart.

While NOAA has accuracy standards for each step in the data collection and chart production process, much of the depth information found on NOA
regarding the accuracy of electronic charts, click http://www.nauticalcharts.noaa.gov/mcd/electronic_accuracy.html

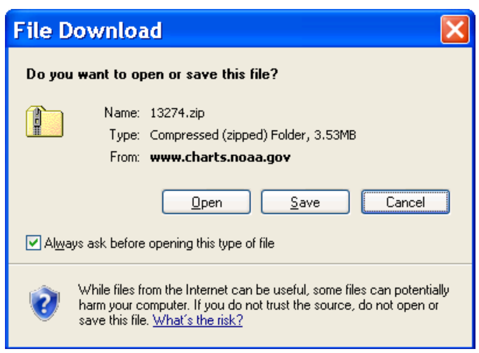
5. Trademarks and Copyright

"NOAA ®" and the NOAA ® emblem are registered trademarks of the National Oceanic and Atmospheric Administration.

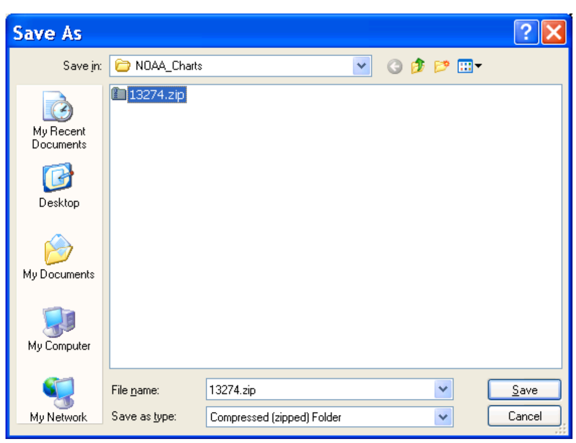
NOAA RNC™ is a trademark of the National Oceanic and Atmospheric Administration.

Click OK to download 13274.zip

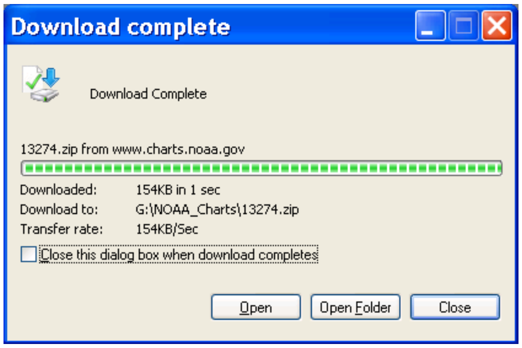
Save the compressed chart to your local computer/server (press **Save**).



Make sure to save it to the designated directory (press **Save**).



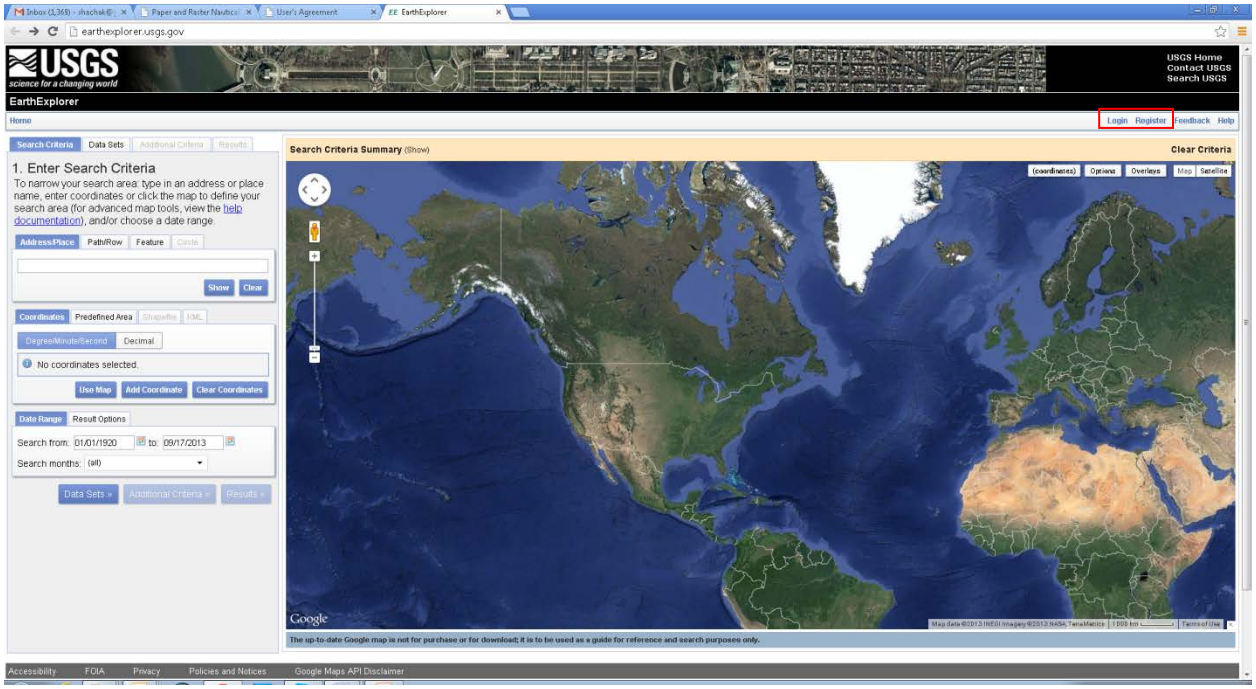
Press **Close** after the **Download complete** window appears



<u>**Note**</u>: The chart is downloaded in a compress format. Make sure to unzip the chart to your data directory.

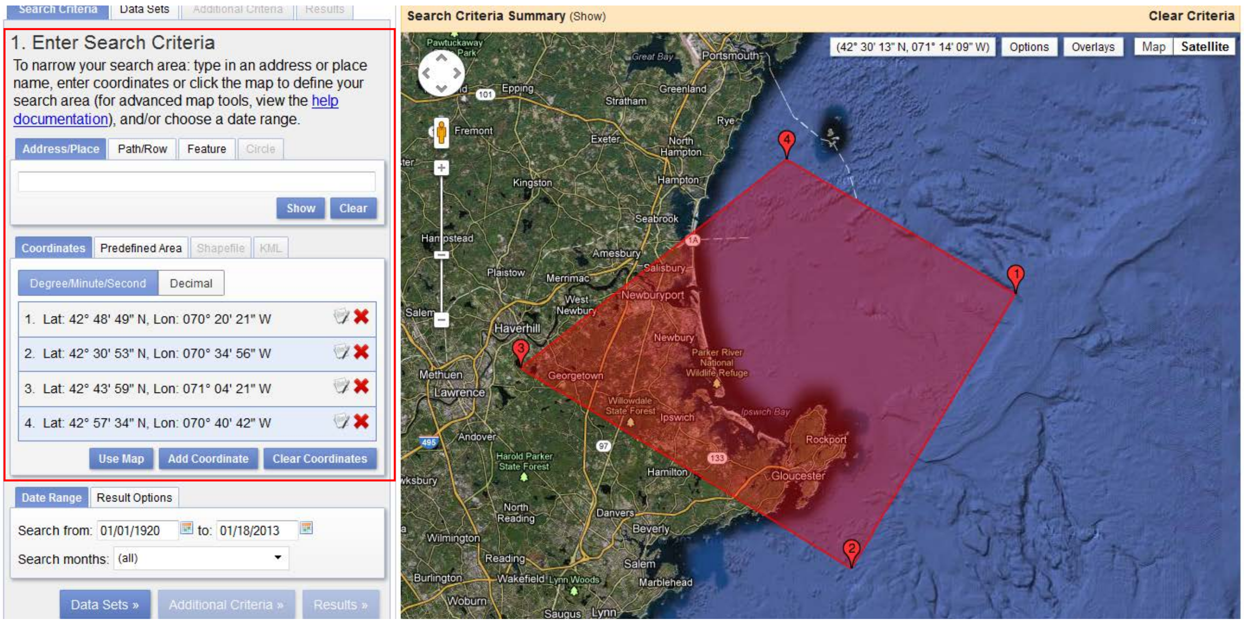
11.1.2 Downloading a Landsat imagery

Open a Web Browser and go to http://earthexplorer.usgs.gov/. Login into your account (the **Login** button is upper right corner of the window).

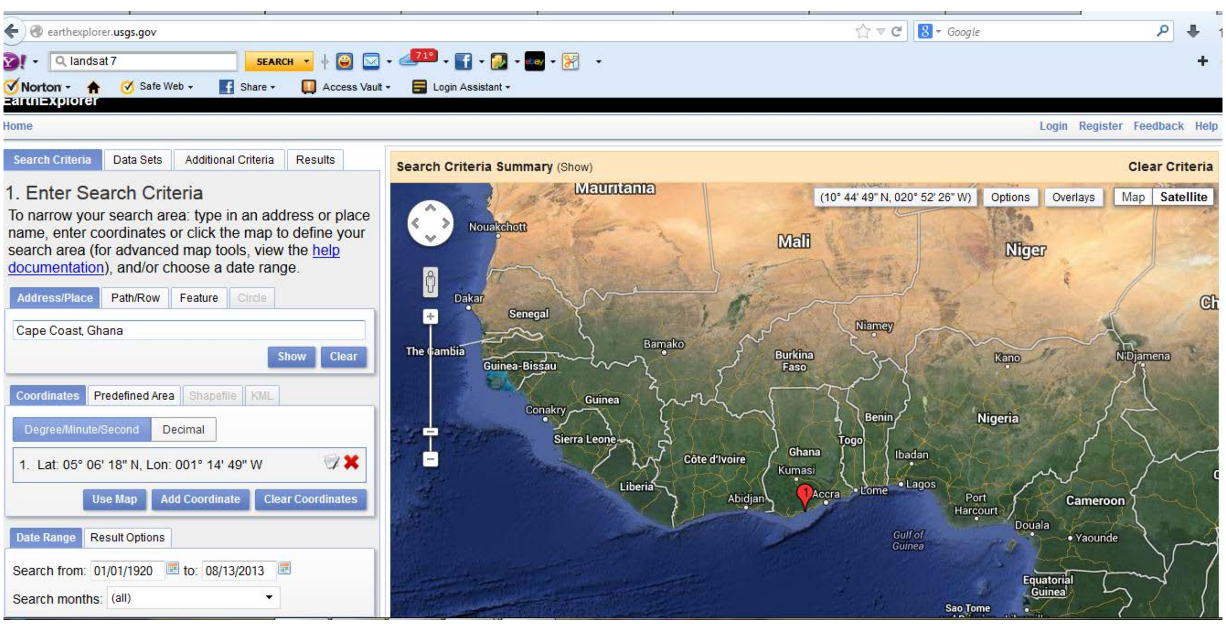


<u>**Note**</u>: The service is free, but the website requires user to login. Users that areaccessing this site for first time, need to register (the **Register** button is upper left corner of the window).

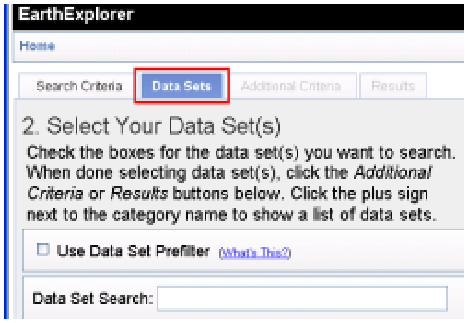
Zoom into the desired region. Create a square around the desired region by clicking on the corners of the area. The polygon vertex coordinates will appear in the **Search Criteria** tab.



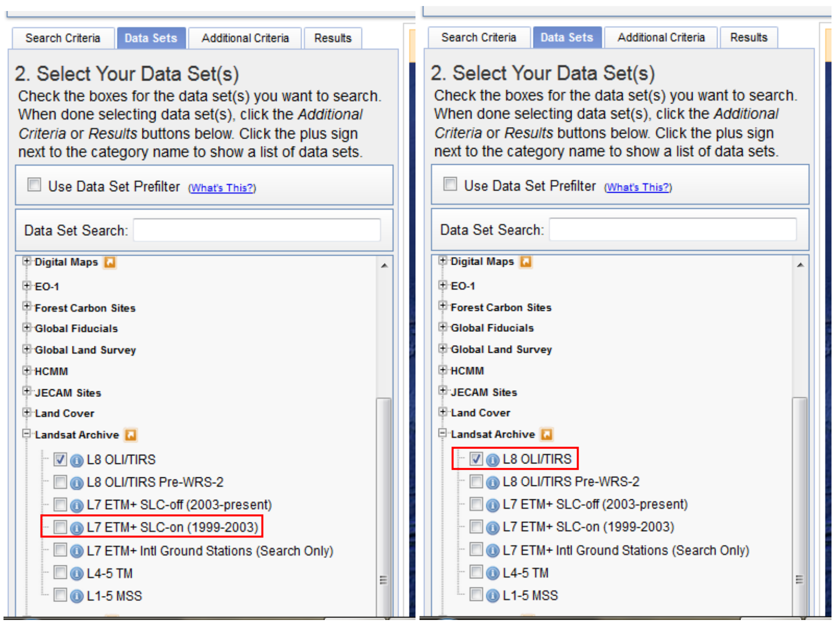
Alternatively, type in a location you are interested and the country in the **Adress/Place** located under the **Search Criteria** tab and click on **Show**. Select the site of interst from the a list of possible sites with similar names and the loction will be shown on the map



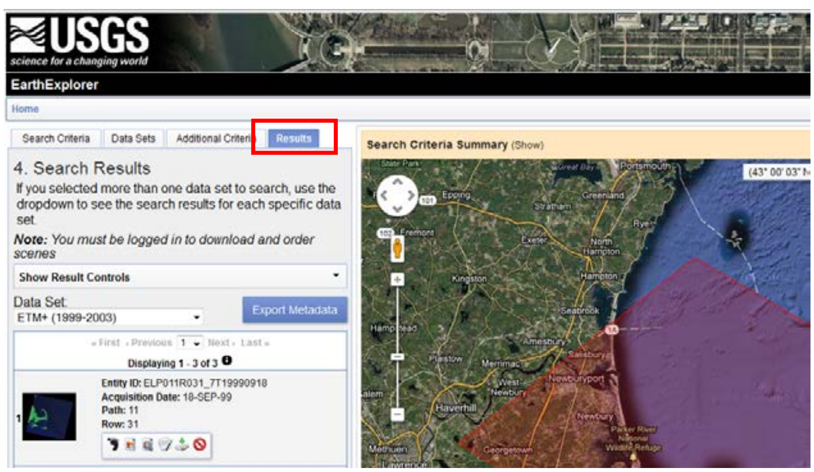
On the top left side of the screen, click on the ***Data Sets*** tab.



Expand the **Landsat Archive** and select L7 **ETM+ SLC-on (1999-2003)** for Landsat 7 imagery or **L8 OLI/TIRS** for Landsat 8 imagery.

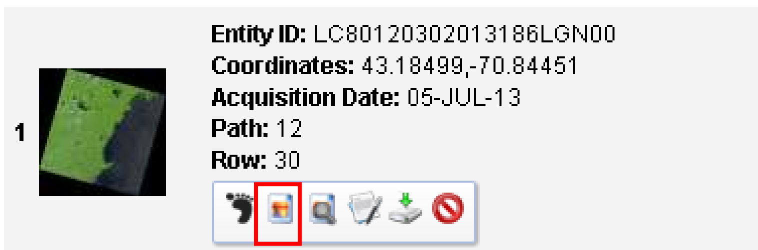


Select the ***Results*** tab.

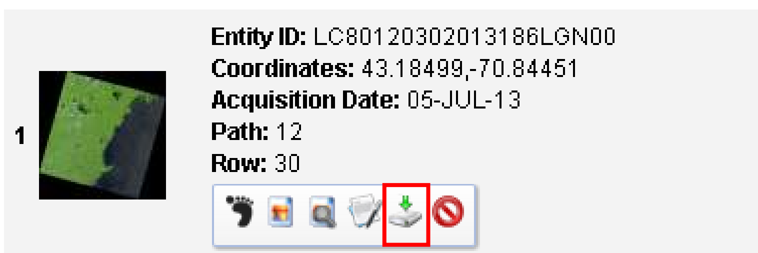


Scroll down and look at the quick view results. Select the download icon of the desired data set.

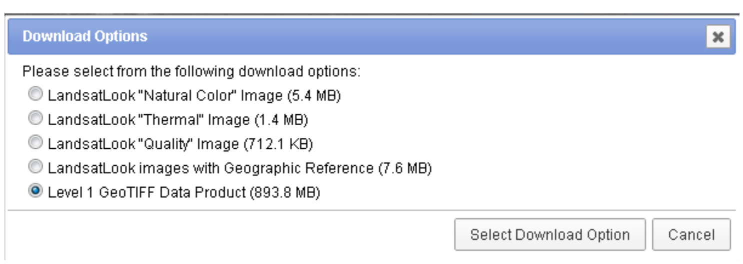
Before secling the image, it is possible to preview the image by slecting ***Show Browse Overlay***.



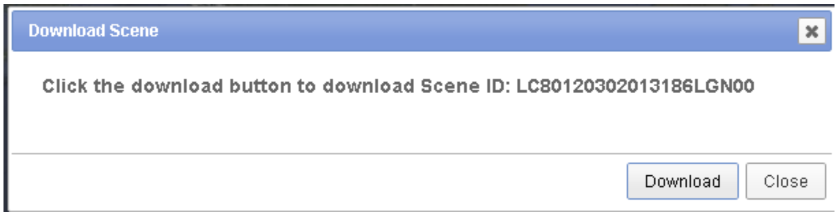
Select ***Download*** to image you are interested to process.



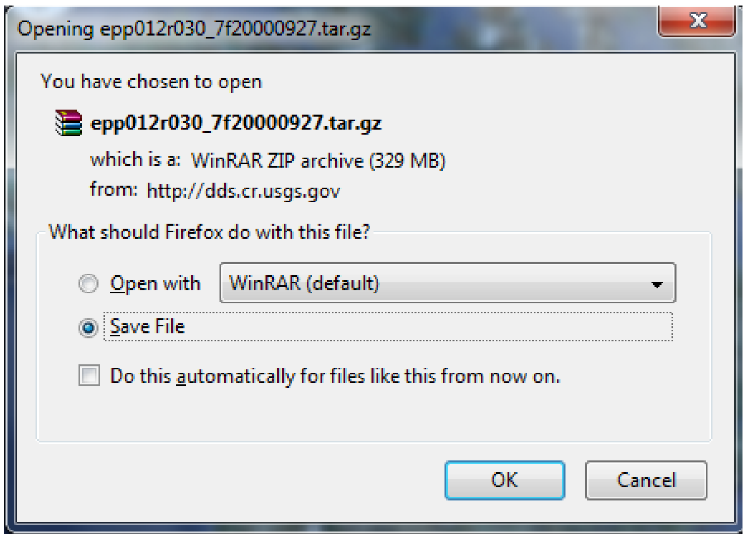
Select ***Level 1 GeoTiff Data Product*** and press ***Select Download Option***.



Press ***Download*** in the **Download Scheme** window.

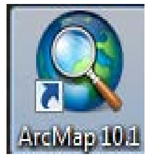


Save the file in your data directory and press ***OK***. Make sure to unzip the chart to your data directory.

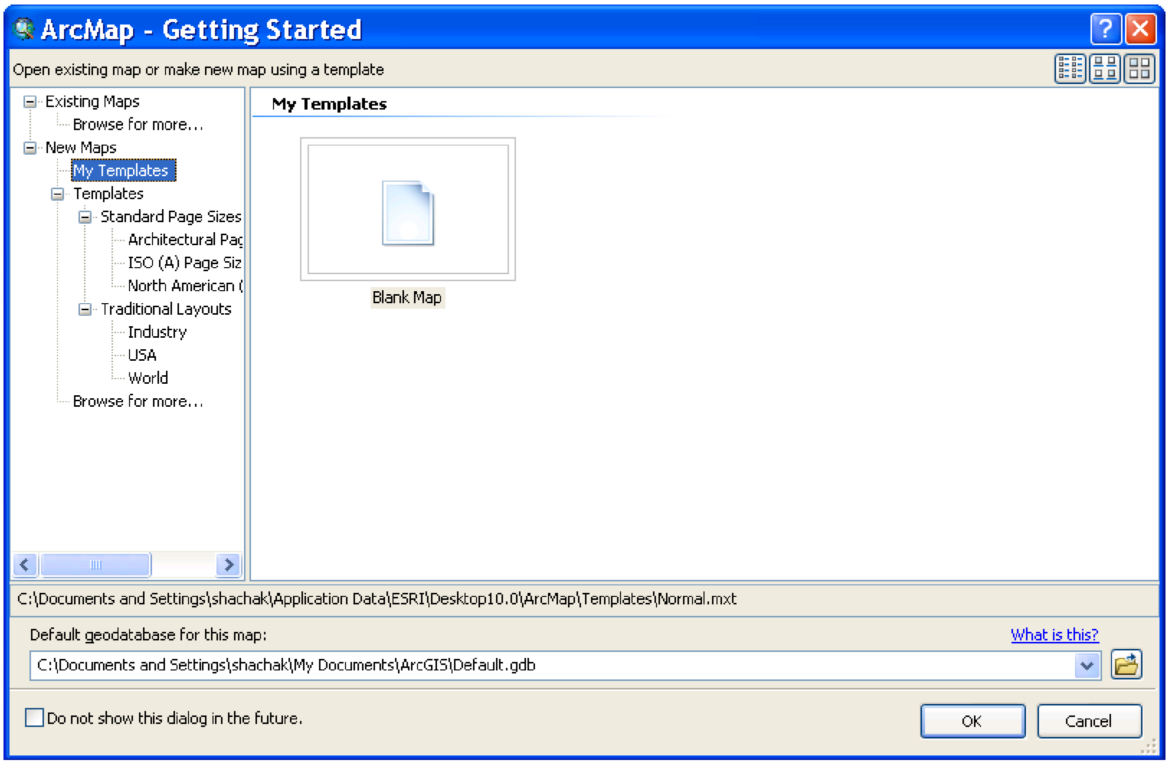


11.1.3 Setting up the GIS environment

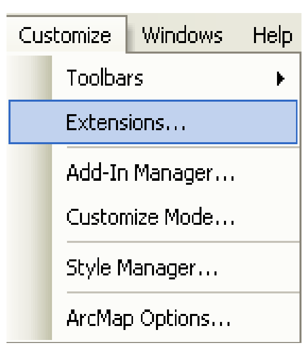
Open ArcMap.



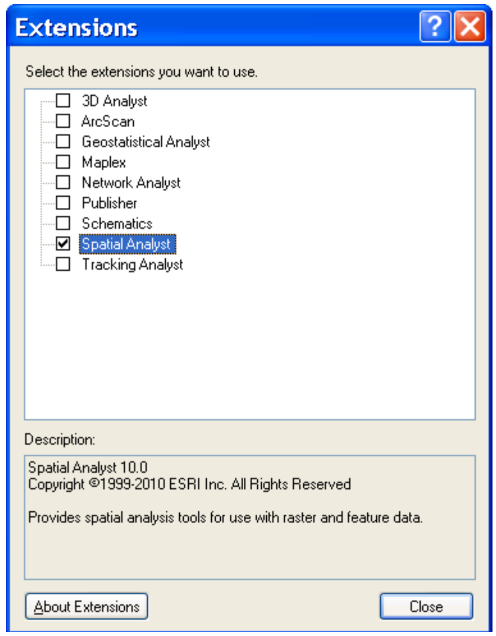
Depending on your set up, you may get the following window. Press ***OK***.



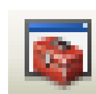
Select ***Extensions…***Under the ***Customize*** Tab.



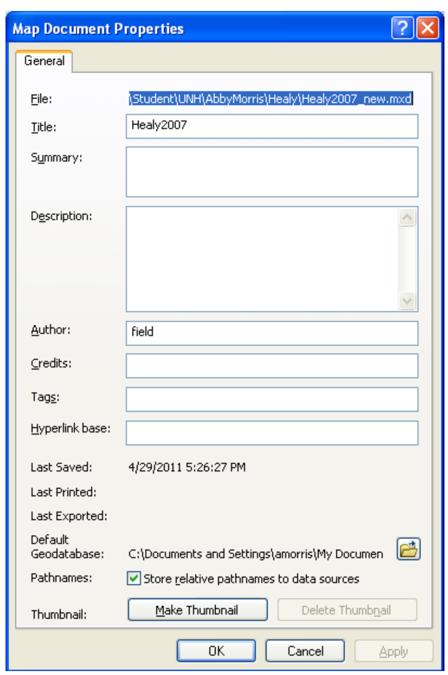
In the ***Extensions*** window, mark **Spatial Analyst**. Press ***Close.***



Activate the **Toolbox** by clicking on the icon in the upper toolbar.



Click on file and open Map Document Properties. Check the box for **Store relative pathnames** to data sources. Click OK.



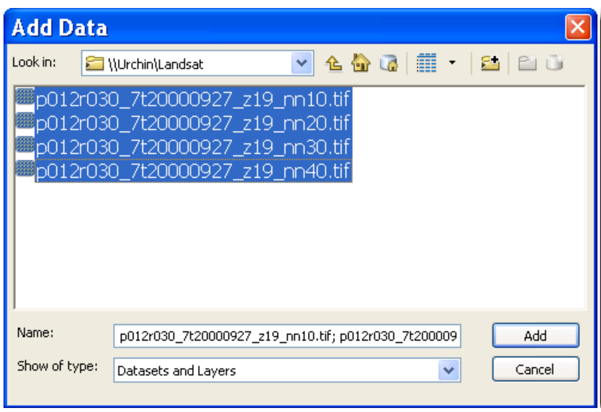
11.2 Loading the datasets in ArcMap

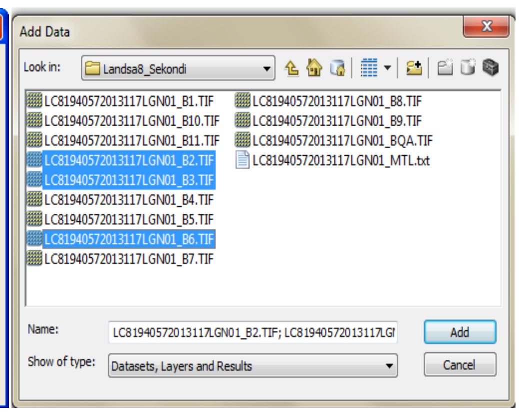
11.2.1 Satellite imagery

Note: Landsat imagery is typically download as separate imagery files and not as a RGB image.

Load the satellite data to ArcMap by selecting ***Add Data***…

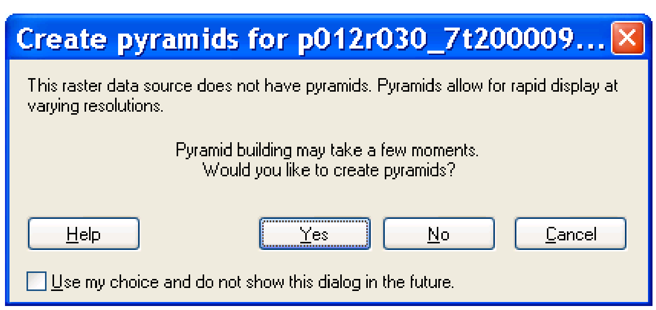
Navigate to your directory and select the satellite image in the **Add Data** window and select ***Add*** button. The Blue, Green and Infrared band in Landsat 7 are numbers as: *_nn10.tif, *_nn20.tif, and *_nn40.tif, respectively. The Blue, Green and Infrared band in Landsat 8 are numbers as: *_B2.tif, *_B3.tif, and *_B6.tif, respectively.



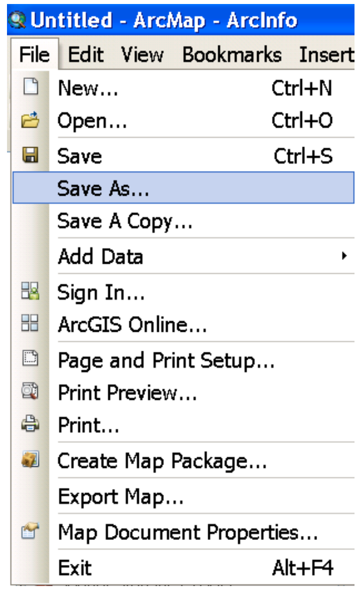


Note: If your imagery is not on your computer, you will need to use the **Connect to Directory** in otder to link your project to your computer.

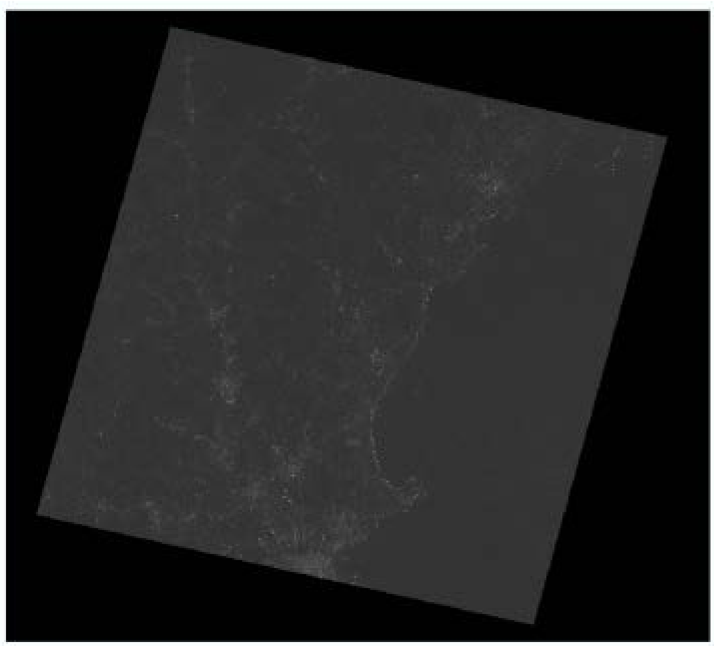
In case you are asked to create pyramids, press ***Yes***.



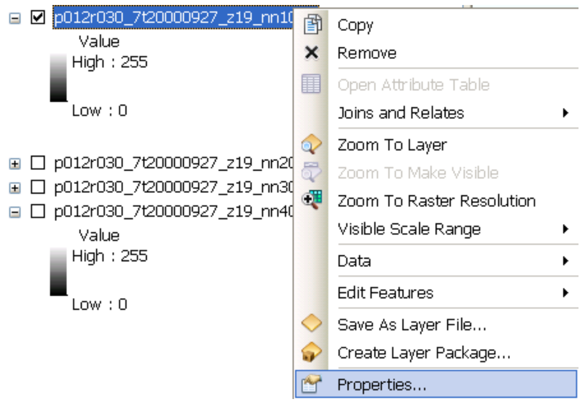
Save your project by selecting **File/ Save As…** .



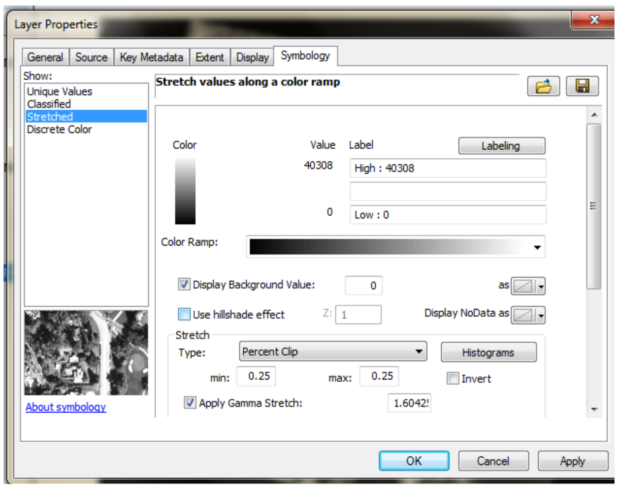
The satellite imagery will be loaded with black background.



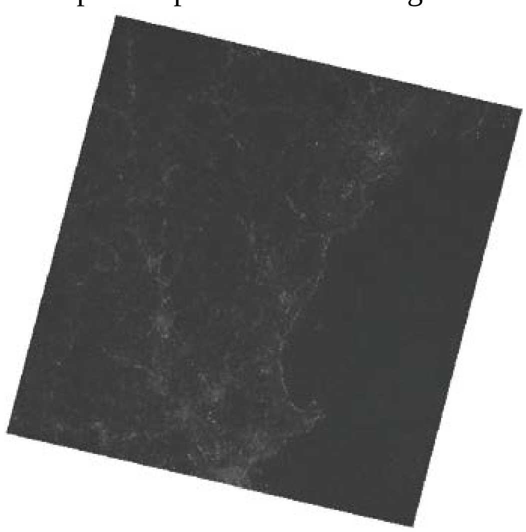
Right click on the image layer and select the ***Properties…*** button.



In the **Symbology** tab, mark the ***Display Background Value*** (in this case black is when the value is 0).



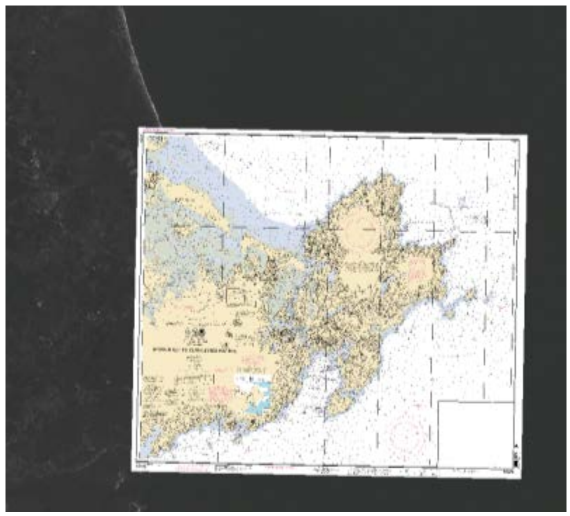
Press the ***Apply*** button and then the ***OK*** button in the Layer Properties window. Now the layer is without the black background. Repeat steps to remove background for the other layers.



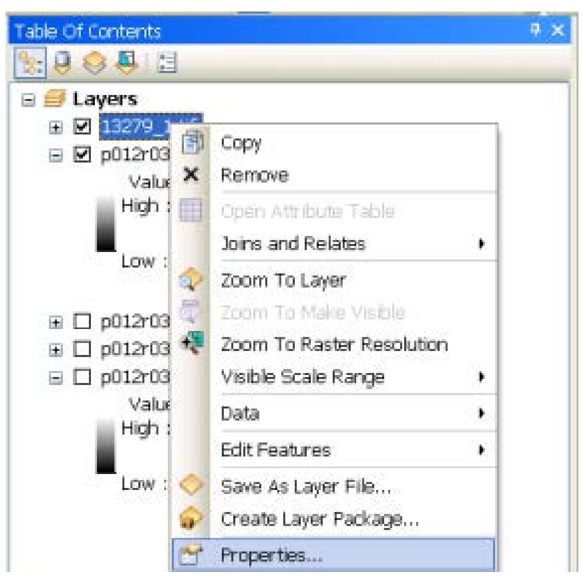
11.2.2 Nautical chart

Load the satellite data to ArcMap by right-clicking on the desired directory and select ***Add Data****…*

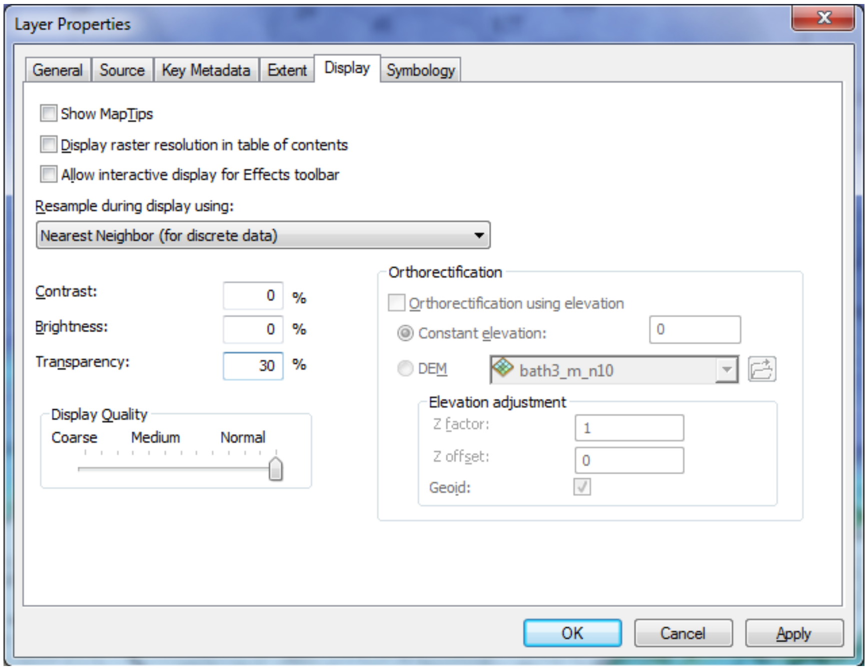
The chart is displayed over the satellite imagery.



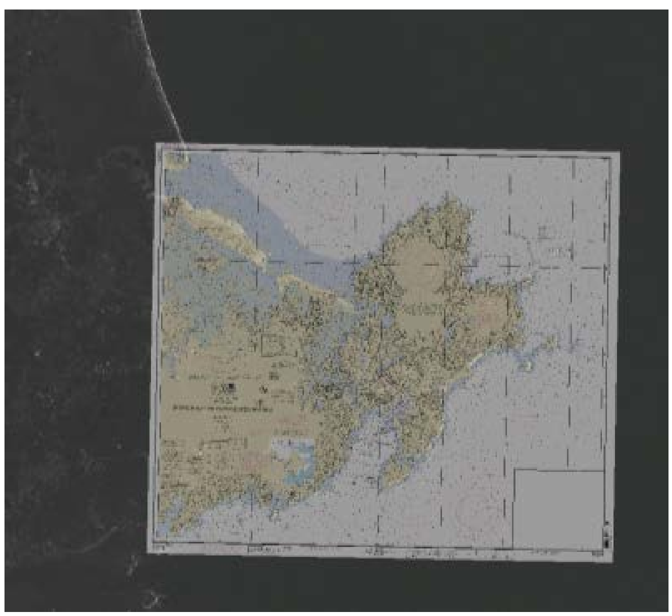
Right click the chart layer and select ***properties***.



Select the ***Display*** tab and set the ***Transparency*** to 30% and press ***OK***.



The chart will appear and also the bathymetry layer below. Make sure the satellite and the chart are registered correctly.



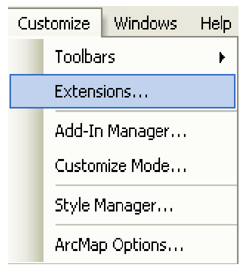
11.3 Water separation and spatial filtering

11.3.1 Calculating the water threshold value

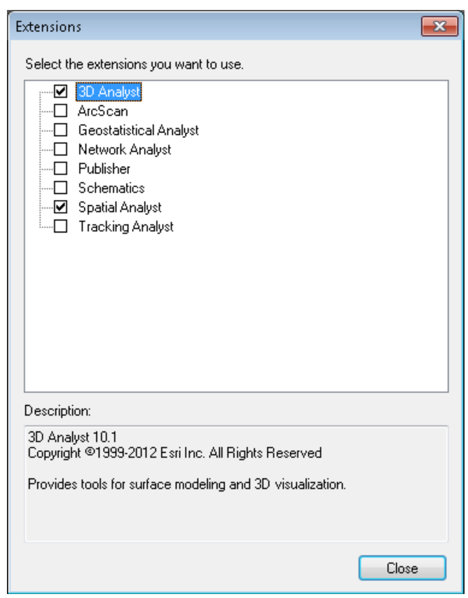
<u>Note</u>: There are several ways to calculate a threshold value for land/water separation. This procedure has options: 1) **Profile,** 2) **Identify,** and 3) **Histogram**. We recommend using the profile option, but we are aware that a **3D Analyst** may with be available with the user's current ArcMap liscence.

<u>**Option 1:** Profile (using **3D Analyst**)</u>

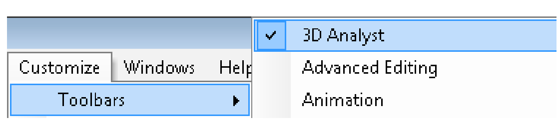
Select ***Extensions…***Under the ***Customize*** Tab.



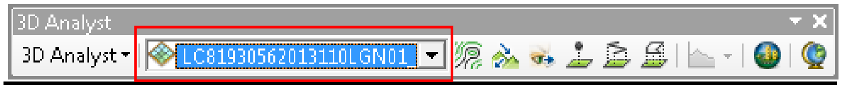
In the ***Extensions*** window, mark **3D Analyst**. Press ***Close.***



Activate the ***3D Analyst*** tool bar under the ***Customize/Toolbars***.

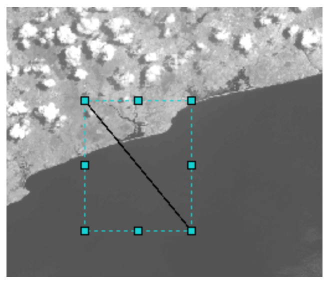


Select the infrared band (*_nn40.tif in Landsat 7 and and *_B6.tif in Landsat 8).

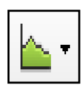


Press the ***Interpolate Line*** icon in the ***3D Analyst*** toolbar.

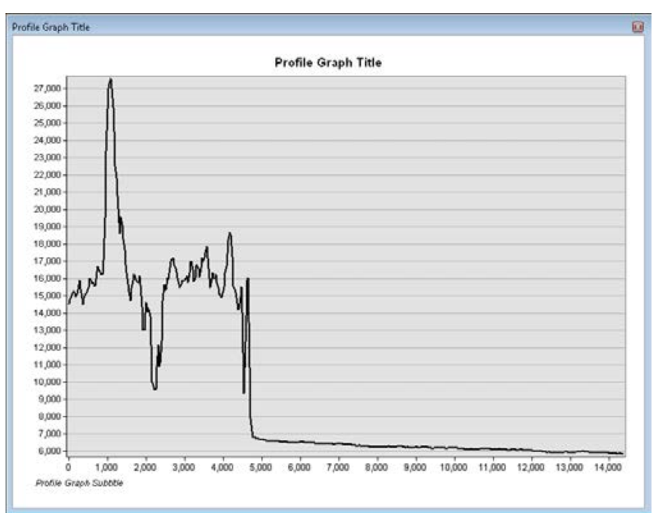
Make sure that the infrared band is visible in the project. Using the left button, draw a line that crosses from land (bright areas) into the water (dark areas). Finalize the line by double-clicking the left button.



After drawing the line the ***Profile Graph*** icon in the ***3D Analyst*** toolbar will be active. Press the ***Profile Graph*** icon.

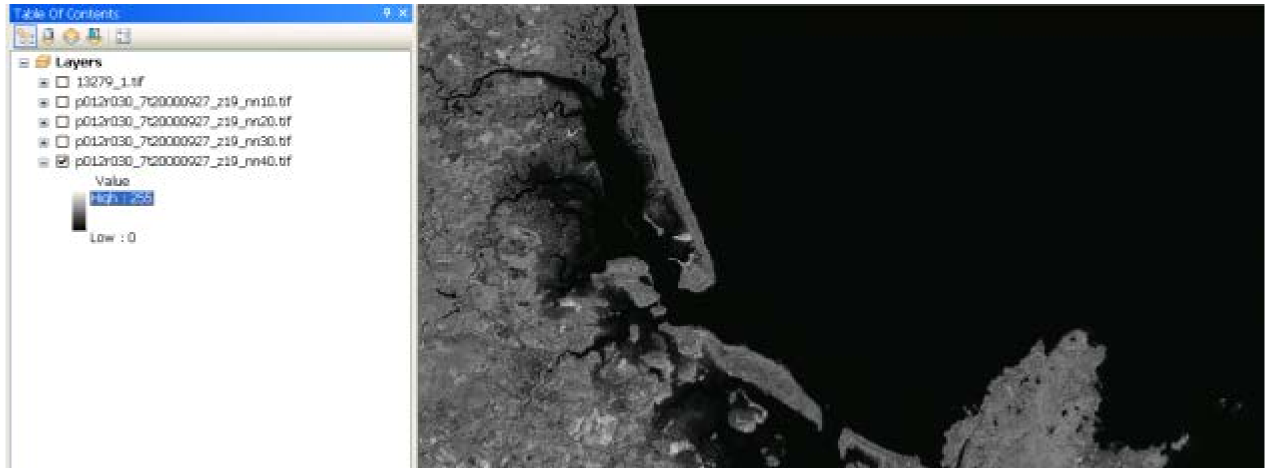


The result will a plot the can be used to extract the land/water threshold. The smouth section with low values represent water, whereas the fluctuation high value areas represents land.In this case the threshold value is around 7000 (Landsat 8 image).



Option 2: Identify

Activate only the infrared band (*_nn40.tif in Landsat 7 and and *_B6.tif in Landsat 8).

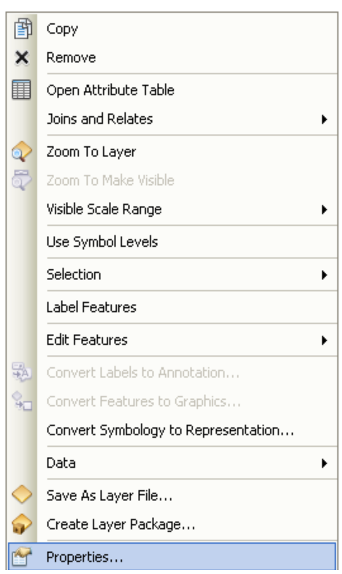


Select the ***Identify*** icon

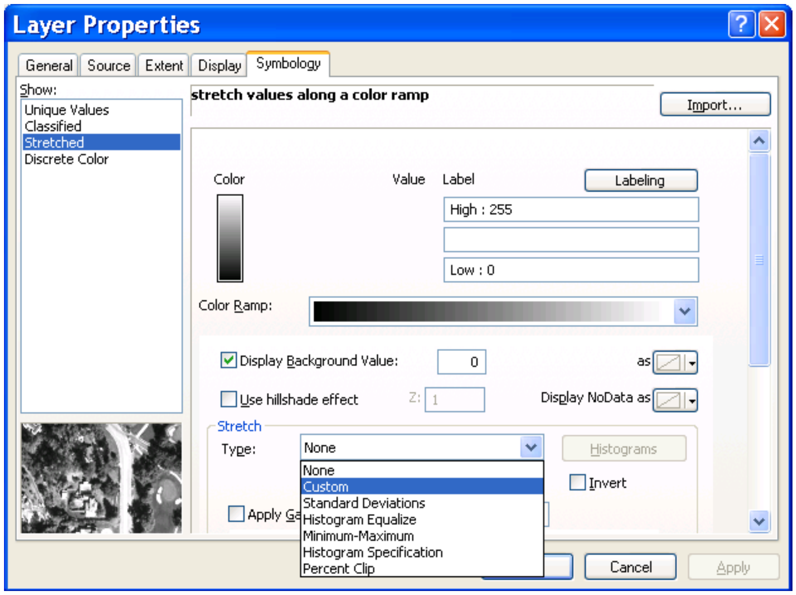
Click on several location over the water and write down the values. Then, click on several location over the land and write down the values. The threshold value should be between the water values and the land values.

Option 3: Histogram

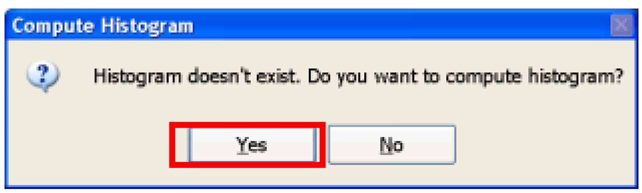
Right click on the infrared layer (in the **Table of Contents**) and select the ***Properties…*** button.



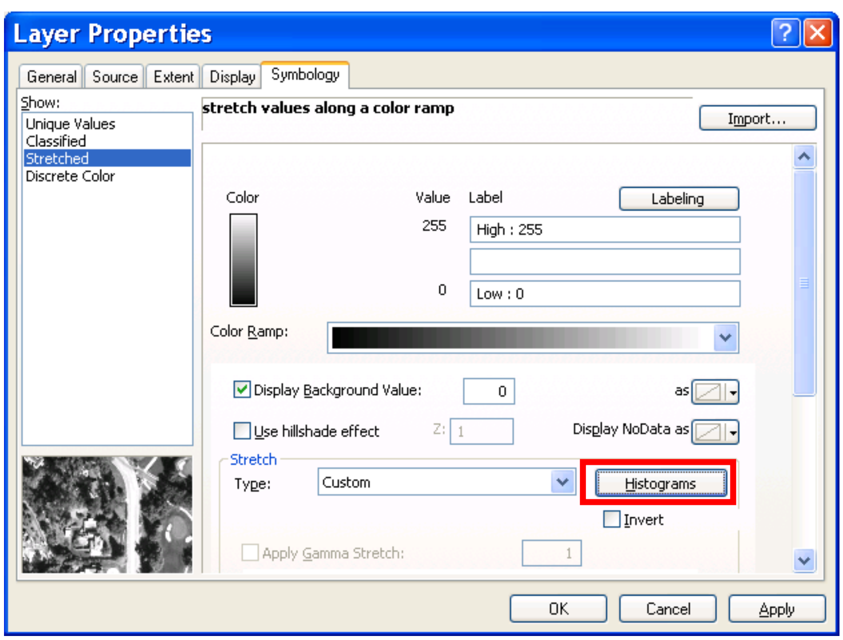
Activate the ***Symbology*** tab. Under ***Stretch***, select **Type: Custom**.



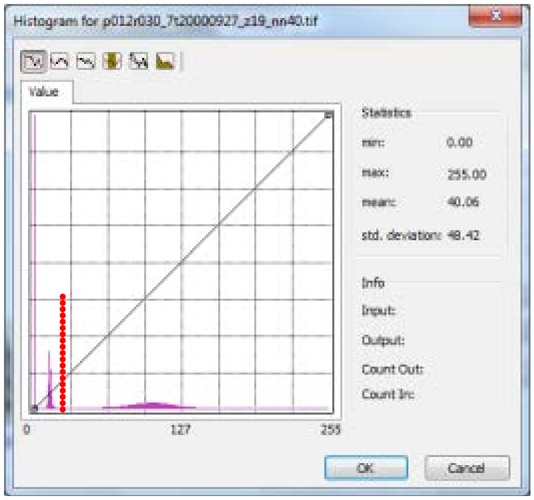
The following message will appear. Press ***Yes***.



Press the ***Histograms*** Button.



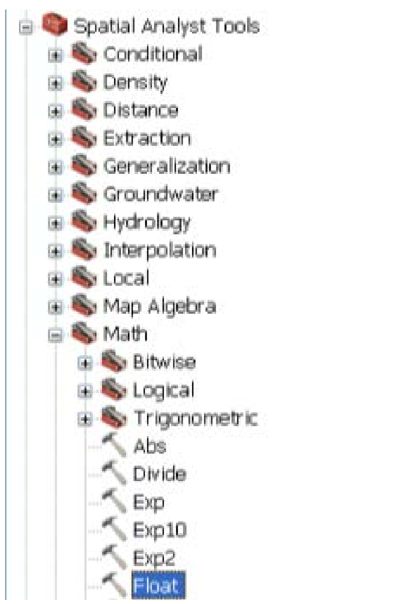
The Infrared Histogram will have two distinct peaks, one represents the land values (i.e., large values) and the other peak represents the water (i.e., values closer to 0).



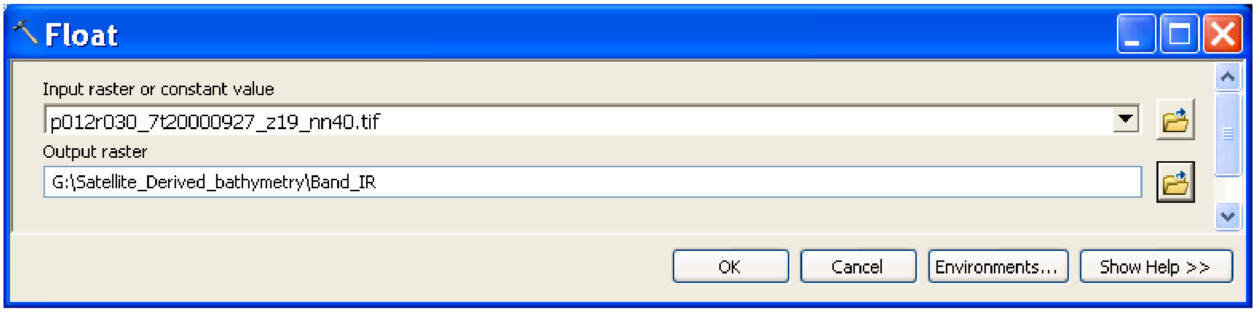
Move the mouse around the graph to determine the threshold by reading the input value. For this layer (Landsat 7 image) the threshold value is 18. This is the threshold value that will be used to generate the water subset mask.

11.3.2 Generating a water subset

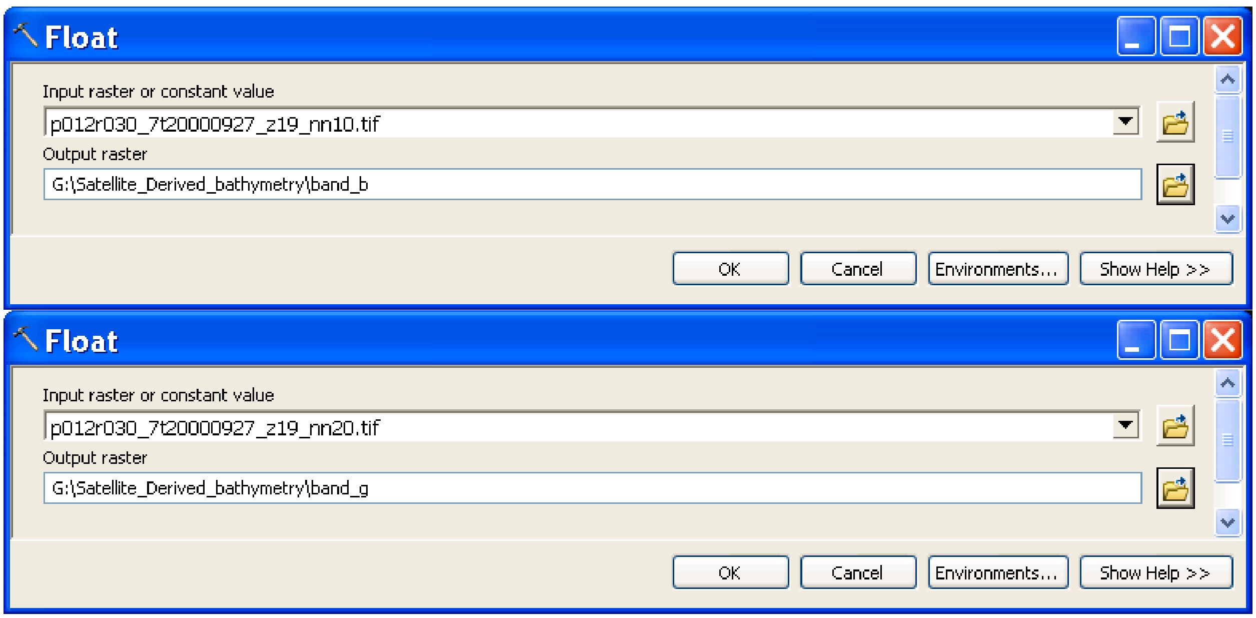
Convert the infrared band into float format, by selecting **Spatial Analyst Tools / Math /Float**from the toolbox



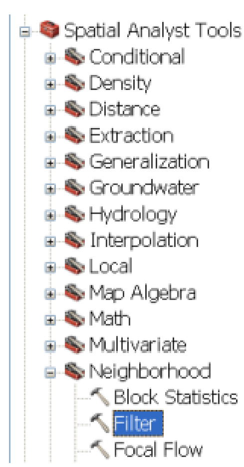
In the **Float** window, select the infrared band (*_nn40.tif in Landsat 7 and and *_B6.tif in Landsat 8) and type in an output raster. Press **OK**.



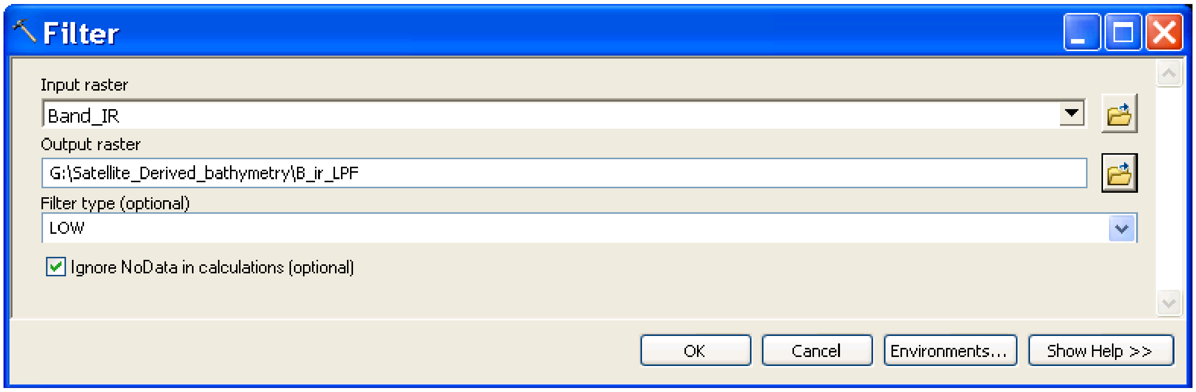
Repeat this step also for the blue (*_nn10.tif in Landsat 7 and and *_B2.tif in Landsat 8) and the green (*_nn20.tif in Landsat 7 and and *_B3.tif in Landsat 8) bands.



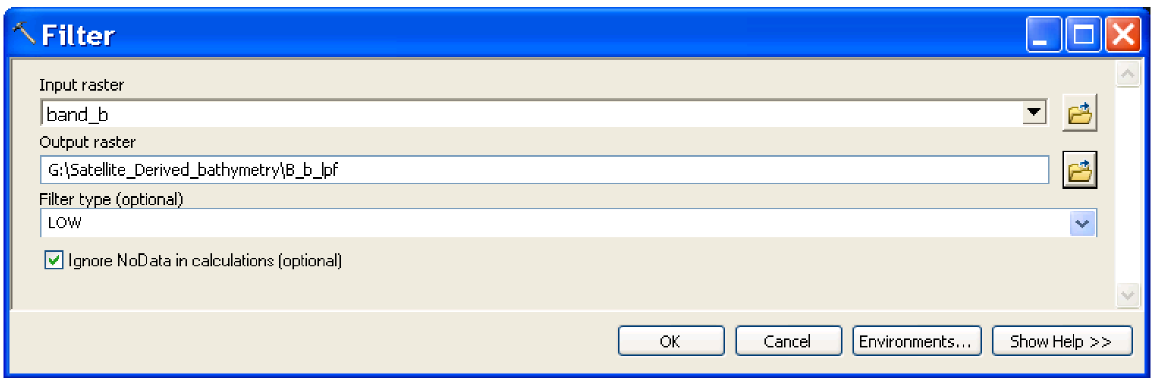
Next, apply a low pass filter by selecting **Spatial Analyst Tools / Neighborhood / Filter**

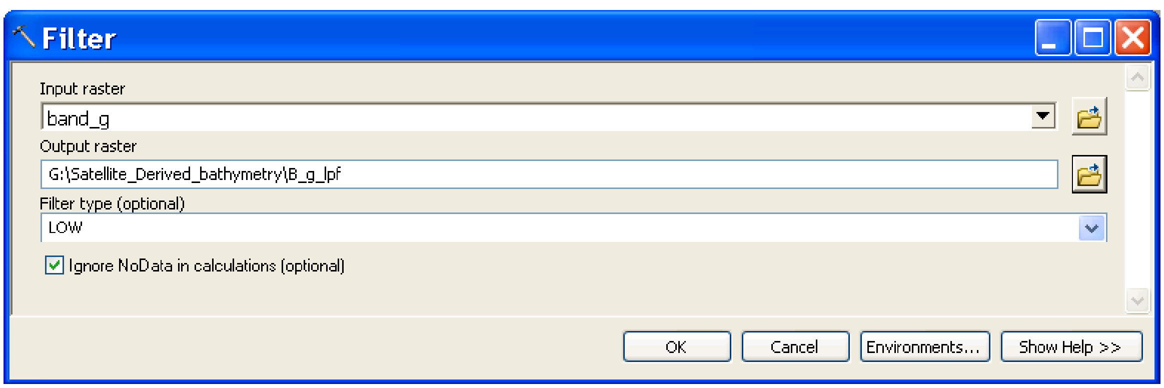


In the **Filter** window select the infrared band in float format and output raster. Also make sure the **Filter type (optional)** is **LOW**. Press **OK**.

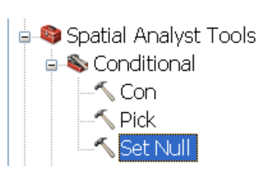


Repeat this step also for the blue and green float files.

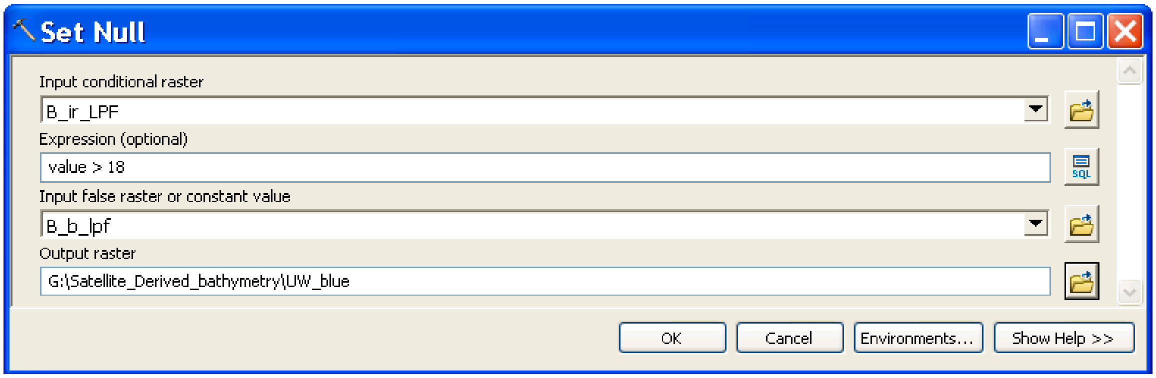




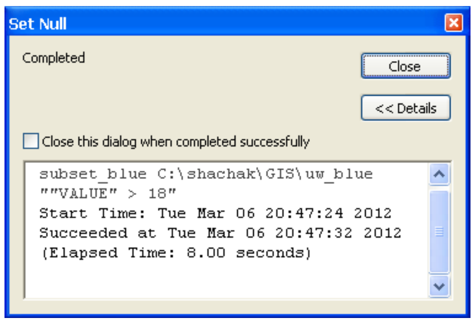
Remove the land from the blue and green imagery, by selecting **Spatial analyst Tools/Conditional/Set Null** in the **Toolbox** window.



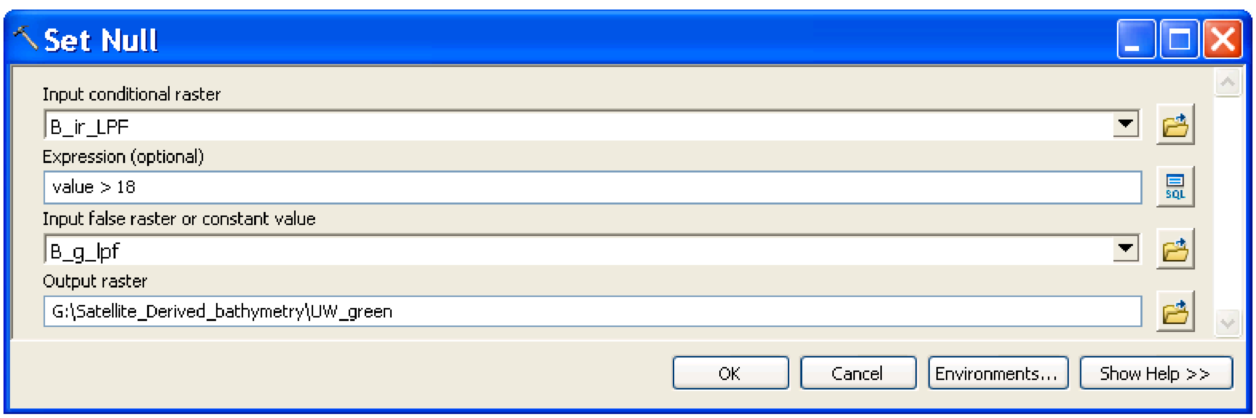
Fill the **Set Null** window as follows for the blue band and press ***OK***. Make sure that the threshold value in the expression is the value calculated in the histogram.



After the **Set Null** process has finished, press ***Close***.



Repeat this step also for the green band



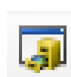
Your water subset layer should contain only areas abive the water and look as follows:

Note: Make sure that the **Display background value** is set to 0 in the layer **Properties.**

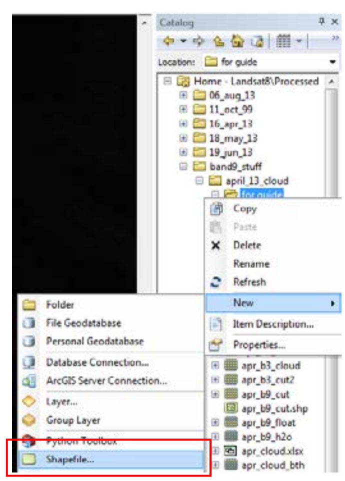
11.4 Glint/cloud correction

<u>**Note**</u>: This step can be performed for imagery from Landsat 7 and Landsat 8. The step is intended to correct radiometric contribution from low altitude clouds and glint from the Blue and Green band.

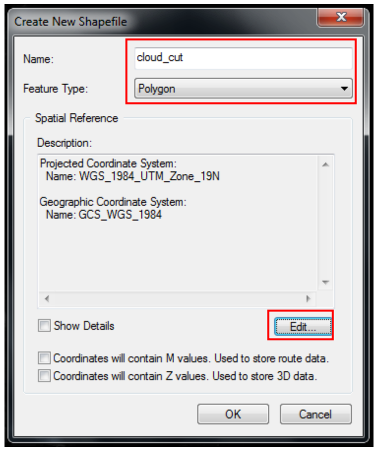
Open the **ArcCatalog**, by clicking on the following button.



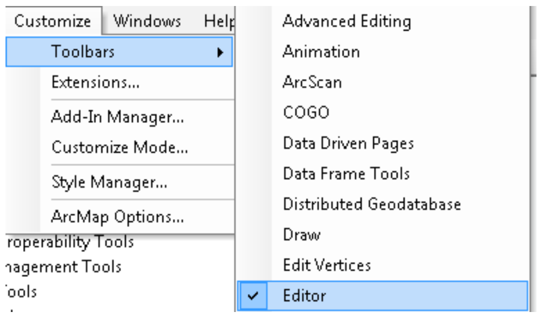
Right click on the folder where the layers are being saved and select ***New*** and then ***Shapefile***.



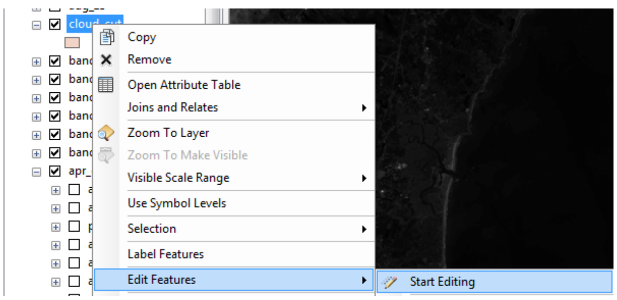
Select ***Polygon*** as the feature type, and name the file **cloud_cut**. Select ***Edit*** to change the coordinate system to the one being utilized for the other layers being worked with. After setting spation reference, press ***OK***.



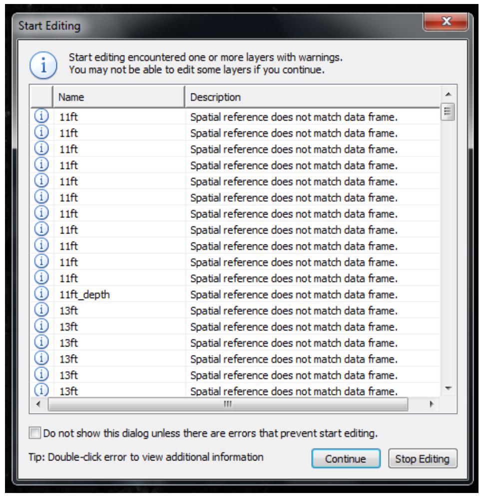
Close **ArcCataolg** and return to **ArcMap**. Activate the ***Editor*** toolbar under ***Customize/Toolbars***.



Right click on the created **cloud_cut** layer and select ***Edit Features*** and then ***Start Editing***.

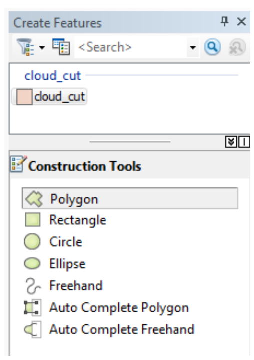


If the following window appears select Continue.

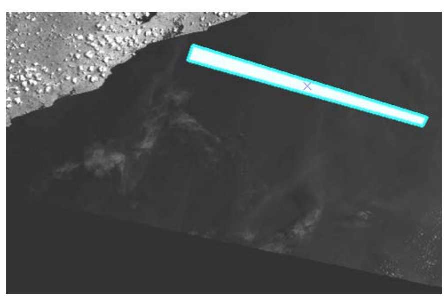


In the ***Editor*** toolbar, press the ***Create Features icon.***

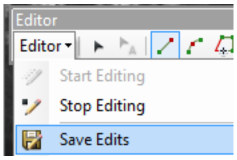
A **Create Features** window will open. Select the "**cloud_cut**" feature and then select ***polygon.***



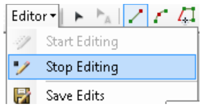
Create a polygon over the infrared layer, which contains both dark sections and bright sections. Create a polygon that is narrow and crosses over the dark areas in the water. To create the polygon, click once at each vertice and to finish double click at the last vertice. **MAKE SURE THIS POLYGON DOES NOT COVER ANY LAND.**



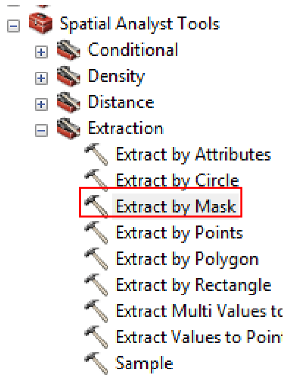
After creating the desired polygon, select ***Save Edits*** in the ***Editor*** toolbar.



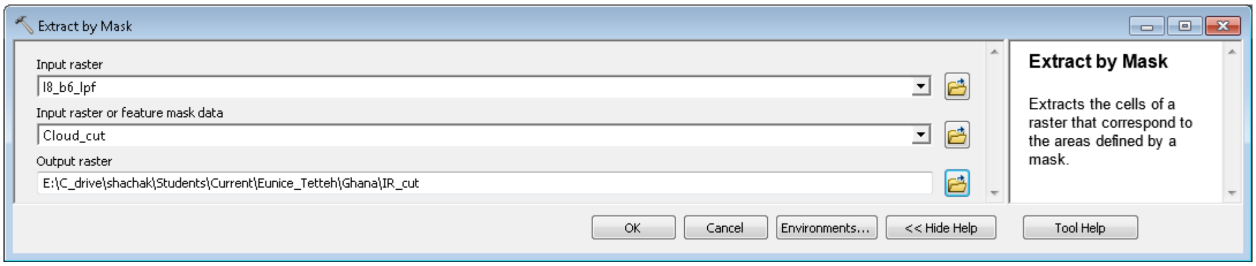
Then select ***Stop Editing*** in the ***Editor*** toolbar.



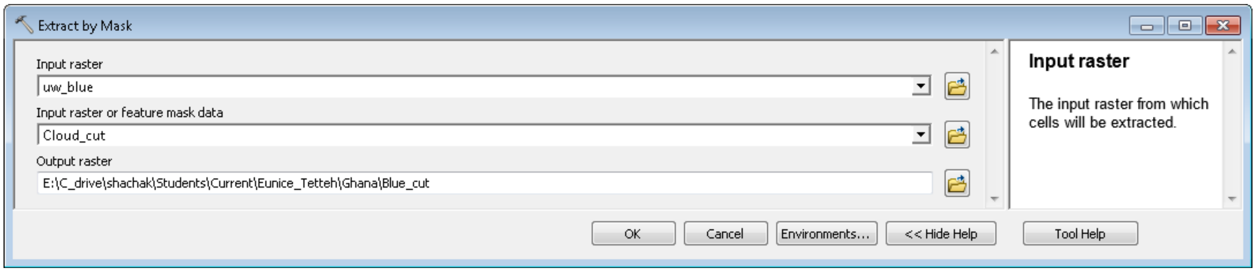
Select **Spatial analyst Tools/Extraction/Extract by Mask** in the **Toolbox** window.

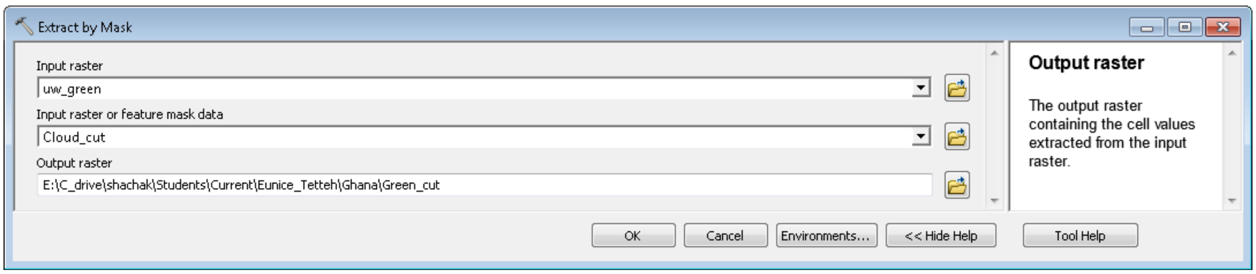


In the **Extract by Mask** window, Input the infrared layer as the raster layer and select ***cloud_cut*** as the mask. Save the output layer as ***IR_cut*** and press ***OK***.

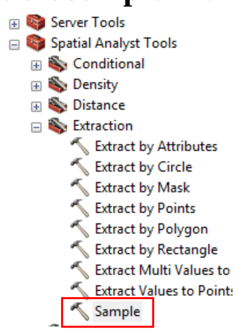


Repeat the ***Extract by Mask*** command with the blue and green layers.

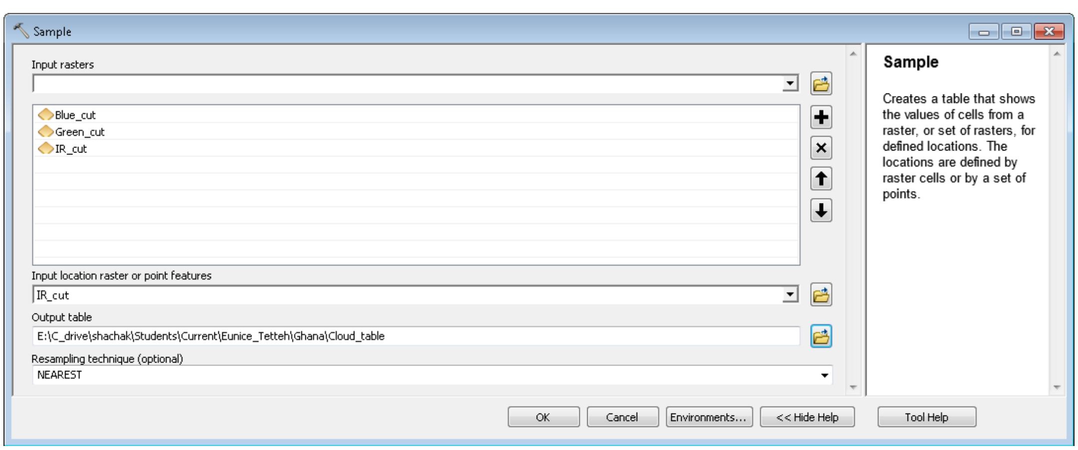




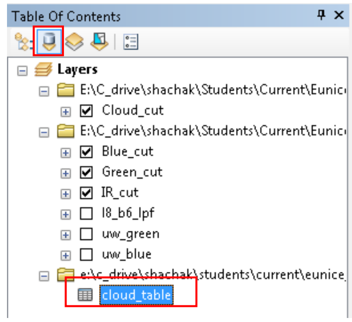
Select **Spatial analyst Tools/Extraction/Sample** in the **Toolbox** window.

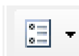


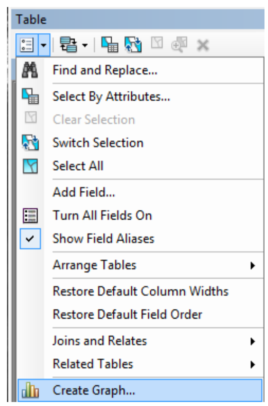
Insert into the **Sample** window the ***Blue_cut, Green_cut,*** and ***IR_cut*** as input rasters. Select the ***IR_cut*** as the input location raster and ***cloud_table*** as the output table. Press ***OK***.



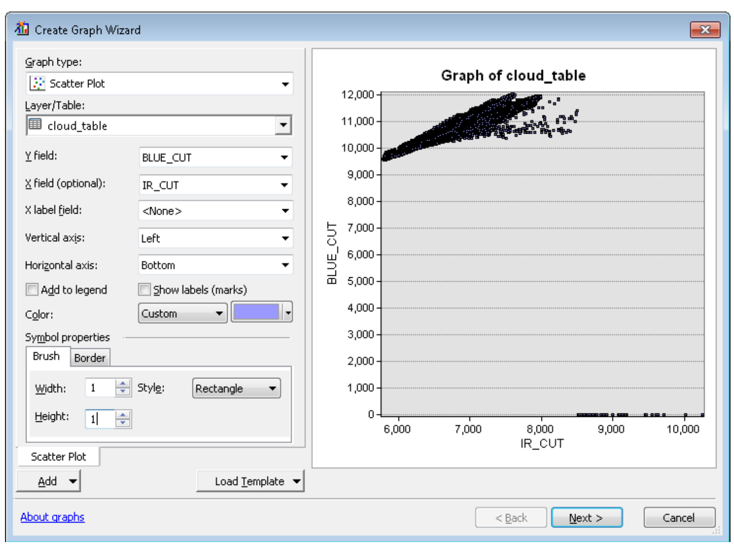
Make sure that your view in the Table of Contents is **List by Source** and then open the ***cloud_table*** by right-click on and select ***Open***.



Select the  button in the **Table** window, and select ***Create Graph***.

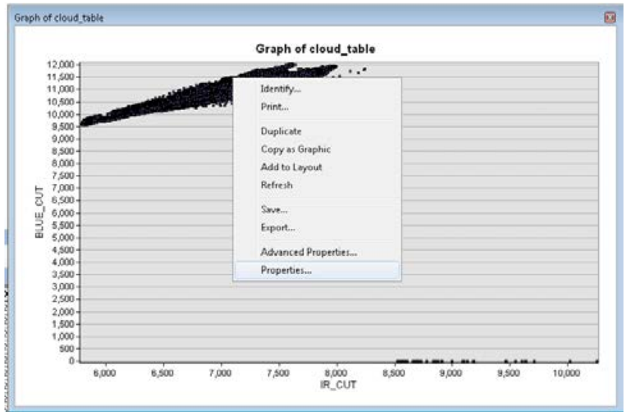


In the **Create Graph Wizard** window, change the graph type to ***Scatter Plot.*** Select ***Blue_cut*** for theY field and ***IR_cut*** for the X field. Press ***Next*** > and then Press ***Finish.***

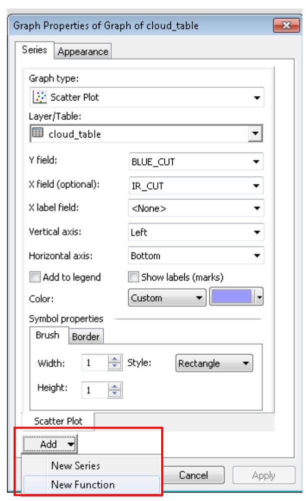


NOTE: It may take the computer more time to respond after each selection, due to amount of points in the dataset.

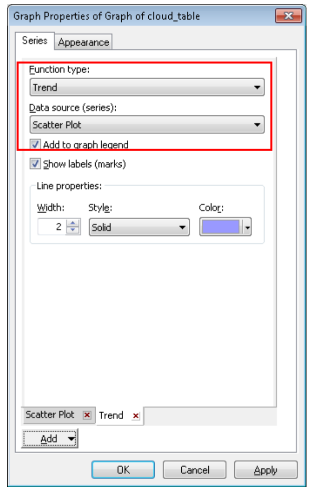
Right click on the graph and select ***Properties...***



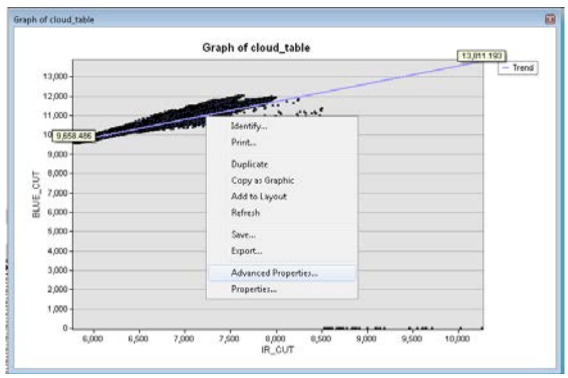
Press the ***Add*** button in the **Graph properties** window and select ***New Function***.



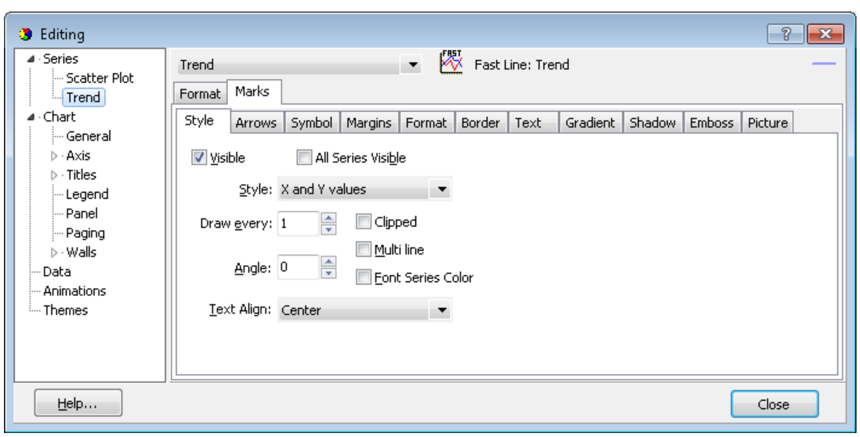
Under the **Series** tab, select ***Scatter Plot*** as the source data and check the ***Show Lables***. Press ***OK***.



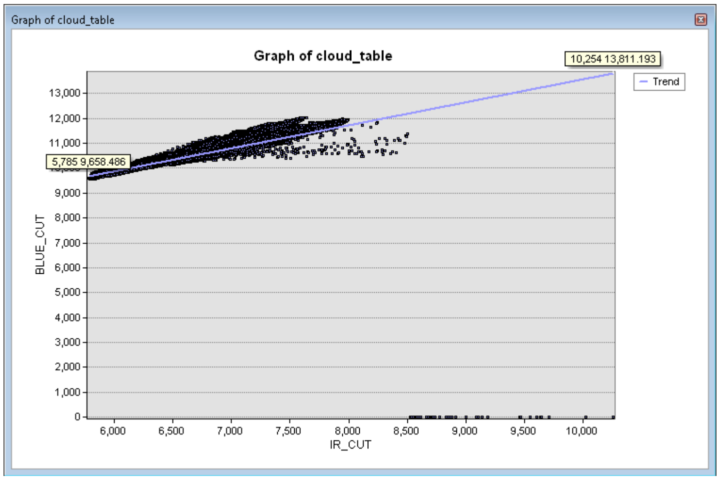
A trend-line and labels have been added to the plot. Right click on the graph and select ***Advanced Properties...***



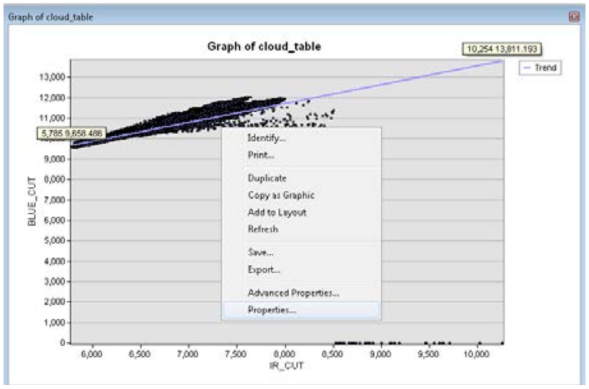
In the Editong window, select Trend and the under the **Mark** tab, select the **Style** tab and change the Style to ***X and Y values***. Press ***Close***.



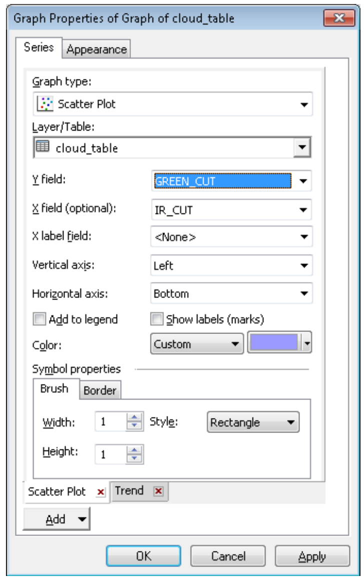
On the graph should be two coordinate points. Record the left point values as the x_1 and y_1 (in this example, x_1=5785 and y_1=9658) and the right point values as the x_2 and y_2 (in this example, x_2=10254 and y_2=13811).



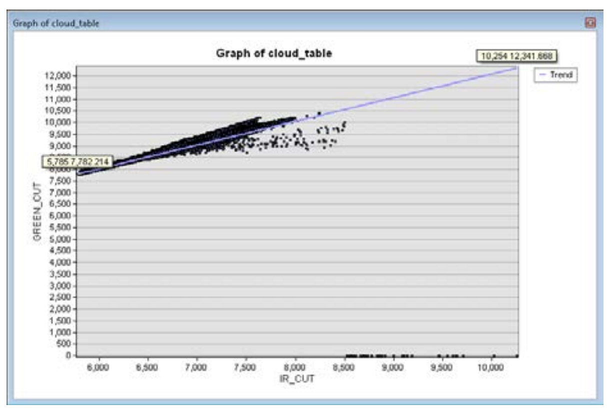
Create a graph and calculate the x_1, y_1, x_2 and y_2 for the ***Green_cut*** layer using the ***IR_cut*** layer. Right click on the graph and select ***Properties...***



On the bottom of the **Graph Properties** window, select the **Scatter Plot** tab. Change the Y field selection to ***Green_cut*** layer using the ***IR_cut*** layer. Press ***OK.***



The Y axis graph has been updated to the green band. In this example, the y_1=7782 and y_2=12342)

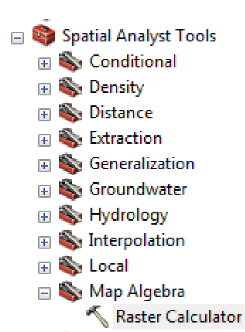


Calculate the slope of the trend for the blue and green layers using the following equation:

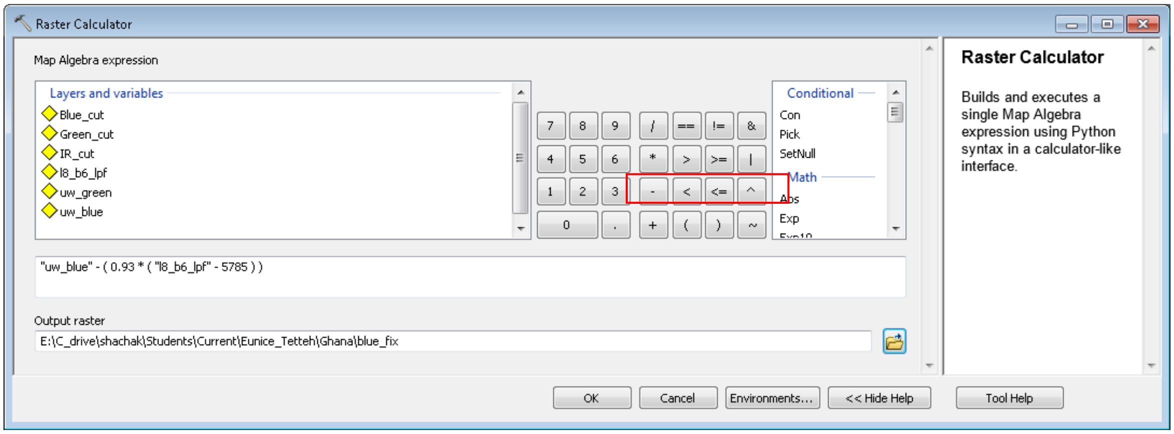
$$\text{Slope} = \frac{y_2 - y_1}{x_2 - x_1}$$

.

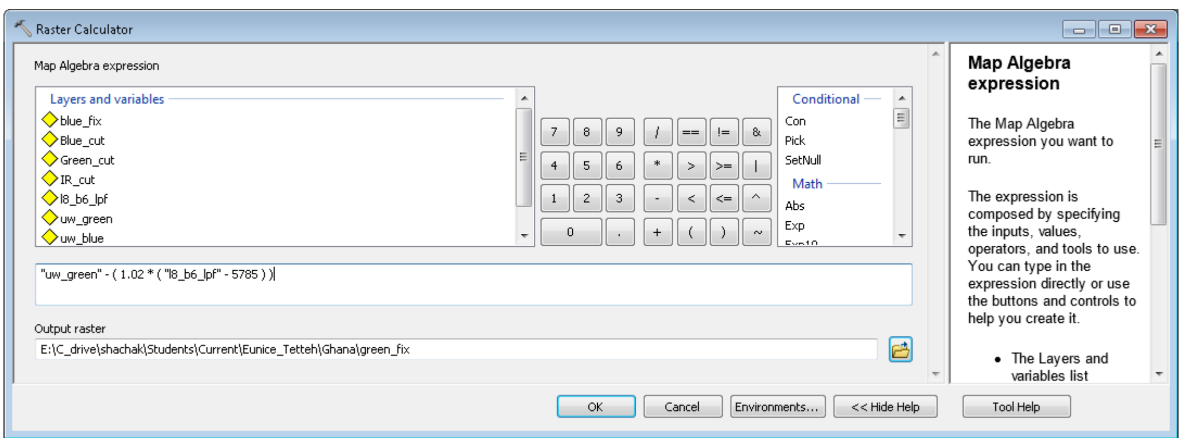
Select **Spatial analyst Tools/Extraction/Raster Calculator** in the **Toolbox** window.



Use the following equation in the raster calculater: ***UW_Blue - Slope * (IR_lpf - X_1).***For this example (slope blue: 0.93), this should be as follows:

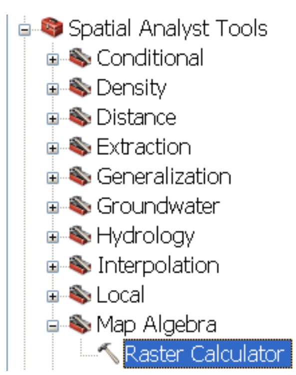


Repeat the calculation for the green band using the following equation: ***UW_Green - Slope * (IR_lpf - X_1).***For this example (slope blue: 1.02):

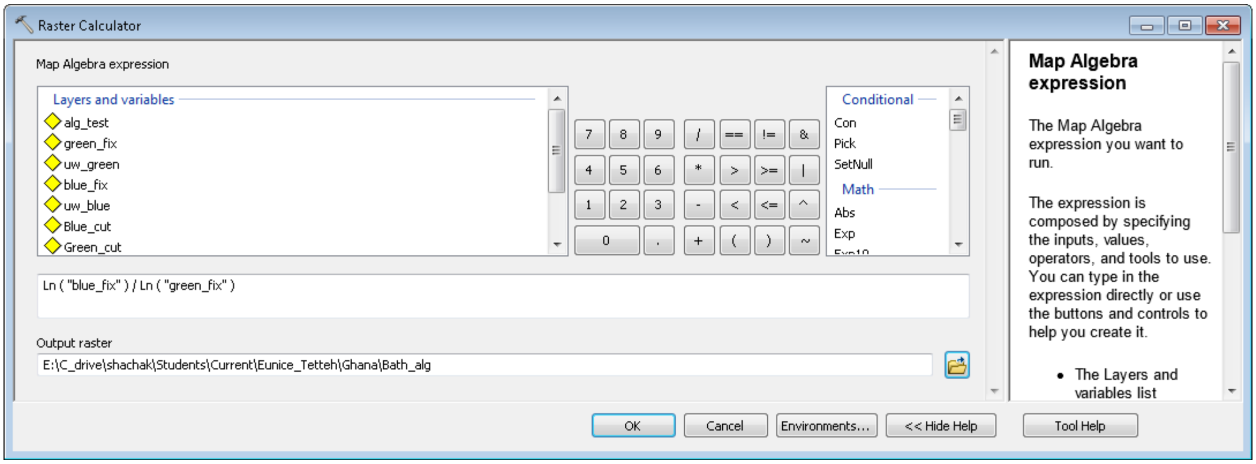


11.5 Applying the bathymetry algorithm

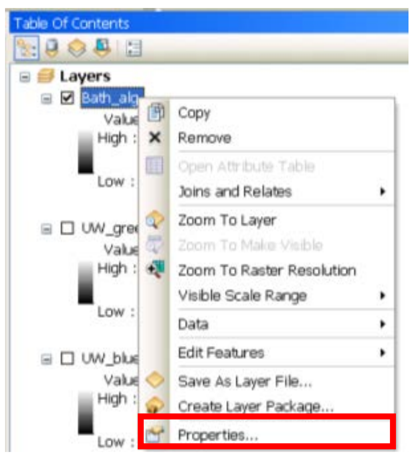
Select **Spatial analyst Tools/Map Algebra/Raster Calculator** in the **Toolbox** window.



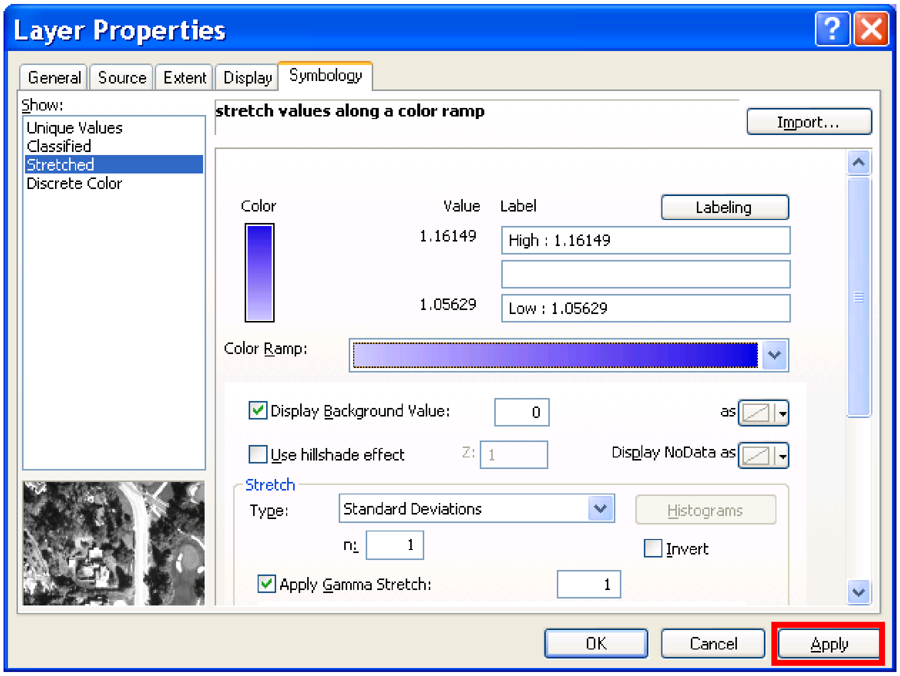
In the Raster Calculator window, write the following command:



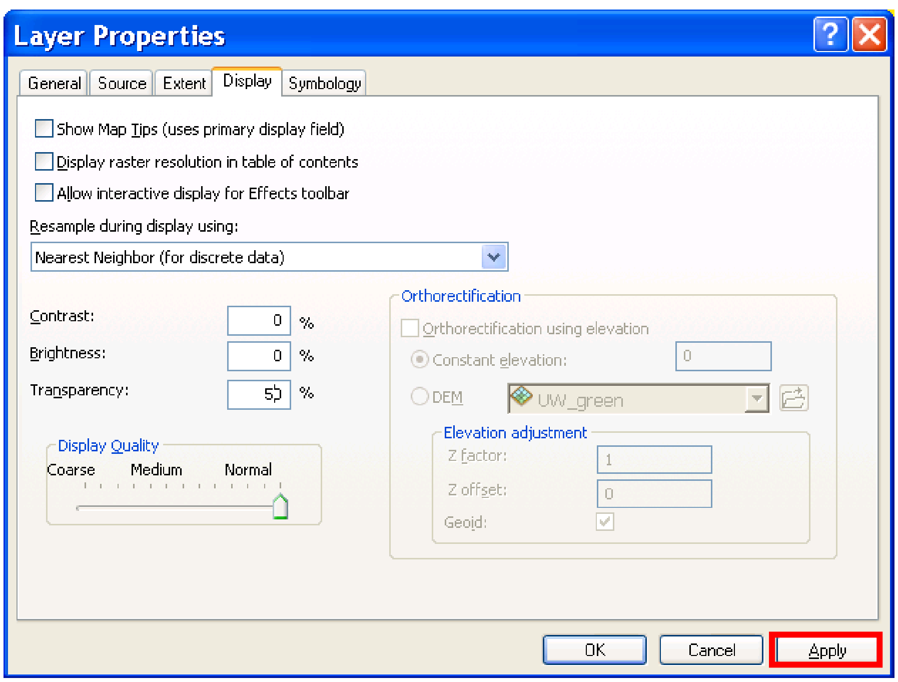
For visual inspection, zoom in to an area of interest (in shallow waters). Select the layer ***properties***



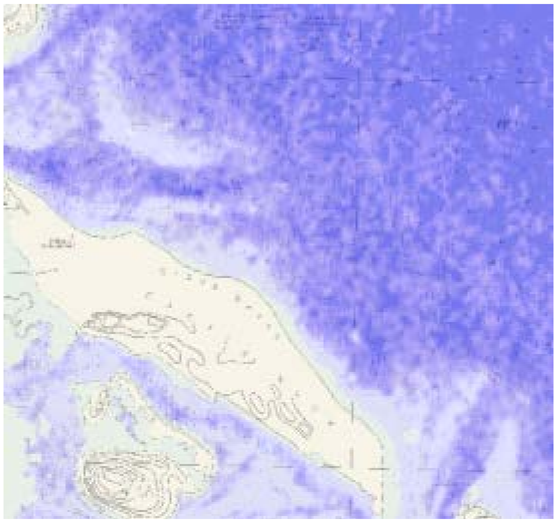
In the **Layer properties** window, select the **Symbology**tab and set the **color ramp and Apply Gamma Stretch: From Current Display Extent**as follows. Press **Apply.**



In the **Layer Properties** window, select the **Display** tab and set the**Transparency** to **50%**. Press OK.



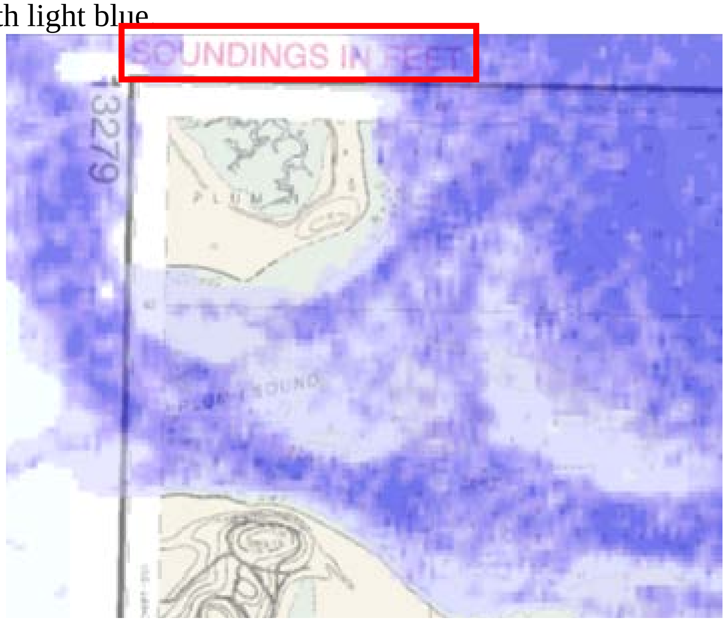
Now, you can compare the algorithm results with the chart



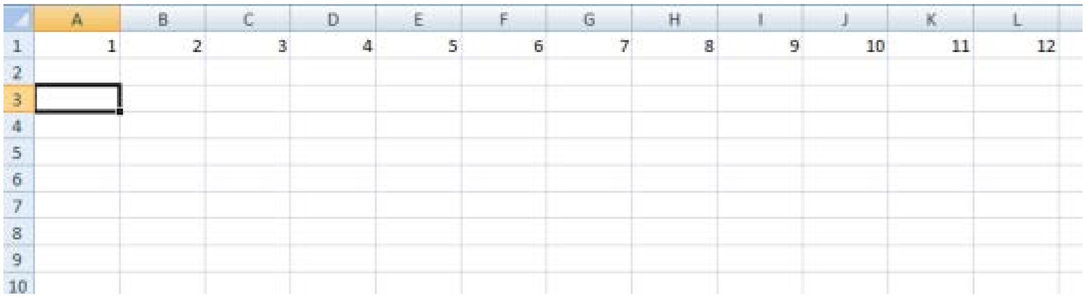
11.6 Vertical referencing and depth of extinction calculation

Note: This is the final step in the procedure and it requires manual work for calculating the gain and offset for referencing the algorithm result to the chart datum. In addition, the effective depth of the bathymetry (i.e., depth of extinction) is calculated.

Identify the units of the soundings and the shallow areas the correlate with depth soundings.In this case, the soundings are in feet. In this example, the soundings are in feet and the shallow areas are marked with light blue.

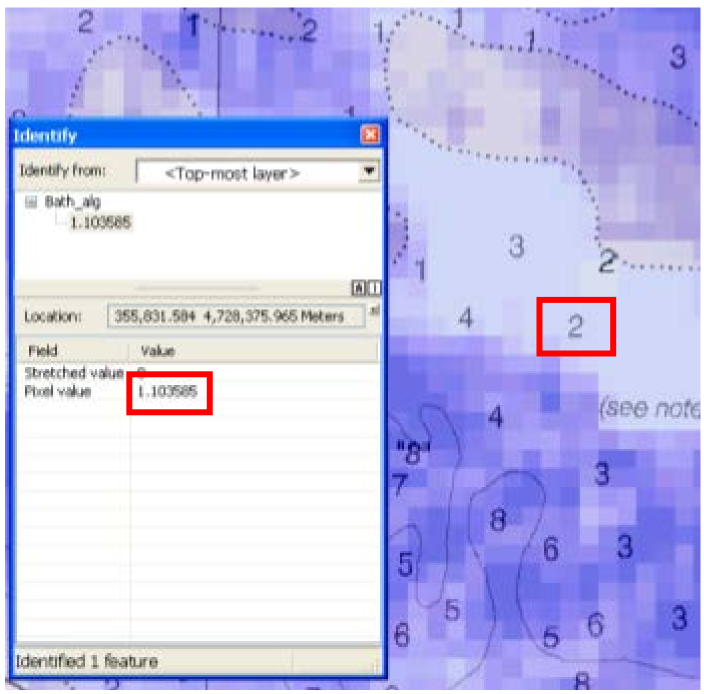


Open an MS Excel. Set up a row of values corresponding to the depth interval. For this data set the intervals are in 1 foot intervals.

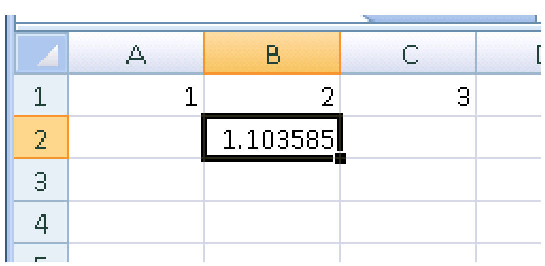


Select the **Identify** tool.

Click on a sounding in the shallow area



Copy the bathymetry value to 4 decimal places in the corresponding depth column in the excel table.



Sample more soundings over shallow areas.Make sure that you have at least 6 samples per depth value.

	A
1	1
2	1.1236
3	1.13119
4	1.1423
5	1.1046
6	1.116
7	1.1155
8	1.1088
9	1.1082

After you finish the shallow depth, proceed to find deeper depth values for the soundings over the deep algorithm results (in this case, over the dark blue areas). Typically 15 m is good in murky waters (e.g., north Atlantic waters) and 30 m in clear water (e.g., Caribbean waters).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	17	20	22
2	1.1236	1.1172	1.1238	1.1338	1.114	1.1143	1.1229	1.137	1.1245	1.1288	1.1289	1.1369	1.1303	1.1308	1.1333	1.131	1.1368	1.1386
3	1.1423	1.134	1.1263	1.1292	1.1263	1.1165	1.1245	1.1336	1.1226	1.1283	1.1359	1.1246	1.1334	1.1322	1.1294	1.138	1.1367	1.1379
4	1.1046	1.0993	1.1141	1.1265	1.1217	1.118	1.1254	1.1263	1.1322	1.131	1.1236	1.1309	1.1324	1.132	1.1327	1.1321	1.1393	1.1358
5	1.116	1.1232	1.1234	1.1258	1.1356	1.1239	1.1245	1.12	1.1294	1.1319	1.122	1.1392	1.134	1.1334	1.1345	1.1381	1.1328	1.1345
6	1.1155	1.1215	1.1205	1.11	1.1214	1.1355	1.1052	1.1295	1.1208	1.1199	1.1222	1.132		1.1362	1.1359	1.1279	1.1327	1.1338
7	1.1088	1.1142	1.1079	1.1162	1.1309	1.1213	1.12	1.1187	1.1322	1.1257	1.1265	1.134		1.1344	1.139	1.1268	1.1302	1.1339
8	1.1082	1.1081	1.126	1.1262	1.1205	1.1107	1.1166	1.1096	1.131	1.1295	1.1321	1.1333		1.137	1.1355	1.1252	1.1336	1.1348
9	1.0966	1.1095	1.1183	1.1314	1.119	1.1263	1.1217	1.1223	1.1328	1.1382	1.135	1.1342		1.1331		1.1337	1.135	1.1357
10	1.1219	1.1159	1.134	1.1144	1.1215	1.1244	1.132	1.1295	1.1263	1.13	1.1371	1.1379		1.132		1.1338	1.139	
11	1.1129	1.1	1.1107	1.1077	1.1216		1.1215	1.1294	1.1259	1.1277	1.1321	1.1073		1.1312		1.1378		
12																		

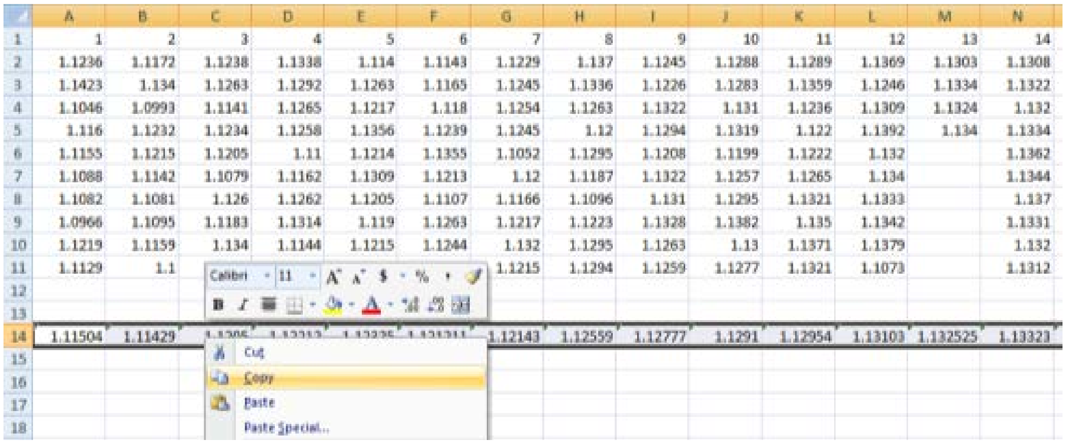
In column A, click on the cell that is two rows below the last value. Type “**=average(**“ and mark the cells you are interested to average.

	A	B	C
1	1	2	3
2	1.1236	1.1036	1.1238
3	1.1423	1.1340	1.1263
4	1.1046	1.0993	1.1141
5	1.1160	1.1232	1.1234
6	1.1155	1.1215	1.1205
7	1.1088	1.1142	1.1079
8	1.1082	1.1081	1.1260
9	1.0966	1.1095	1.1183
10	1.1219	1.1159	1.1340
11	1.1129	1.1000	1.1107
12			
13	=AVERAGE(A2:A11		
14	AVERAGE(number1, [number2		

Sheet1 Sheet2

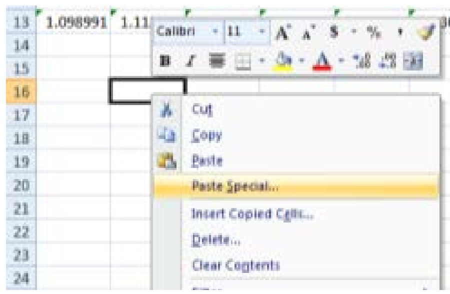
Repeat this average calculation for the other columns. Make sure that all the average values are in the same row.

Select all the average values by left-clicking and marking them all.Right-click on the marked area and select**Copy**.

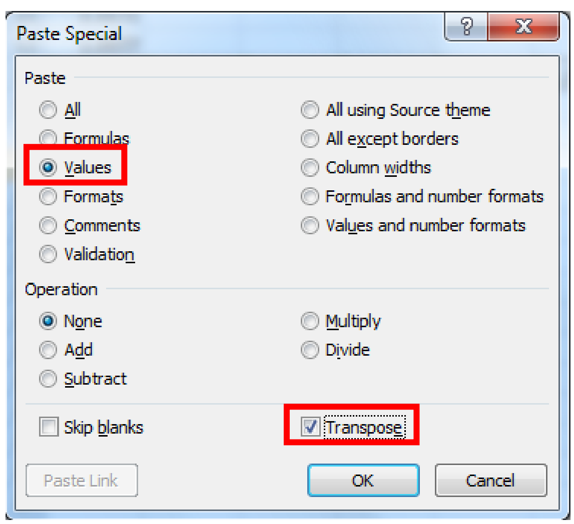


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1.1236	1.1172	1.1238	1.1338	1.114	1.1143	1.1229	1.137	1.1245	1.1288	1.1289	1.1369	1.1303	1.1308
3	1.1423	1.134	1.1263	1.1292	1.1263	1.1165	1.1245	1.1336	1.1226	1.1283	1.1359	1.1246	1.1334	1.1322
4	1.1046	1.0993	1.1141	1.1265	1.1217	1.118	1.1254	1.1263	1.1322	1.131	1.1236	1.1309	1.1324	1.132
5	1.116	1.1232	1.1234	1.1258	1.1356	1.1239	1.1245	1.12	1.1294	1.1319	1.122	1.1392	1.134	1.1334
6	1.1155	1.1215	1.1205	1.11	1.1214	1.1355	1.1052	1.1295	1.1208	1.1199	1.1222	1.132		1.1362
7	1.1088	1.1142	1.1079	1.1162	1.1309	1.1213	1.12	1.1187	1.1322	1.1257	1.1265	1.134		1.1344
8	1.1082	1.1081	1.126	1.1262	1.1205	1.1107	1.1166	1.1096	1.131	1.1295	1.1321	1.1333		1.137
9	1.0966	1.1095	1.1183	1.1314	1.119	1.1263	1.1217	1.1223	1.1328	1.1382	1.135	1.1342		1.1331
10	1.1219	1.1159	1.134	1.1144	1.1215	1.1244	1.132	1.1295	1.1263	1.13	1.1371	1.1379		1.132
11	1.1129	1.1					1.1215	1.1294	1.1259	1.1277	1.1321	1.1073		1.1312
12														
13														
14	1.11504	1.11429	[illegible]	[illegible]	[illegible]	[illegible]	1.12143	1.12559	1.12777	1.1291	1.12954	1.13103	1.132525	1.13323
15														
16														
17														
18														

Select a data box that is below in **Column B**the data, right clickand select **Paste Special**.



In the **Paste Special** window, select **Values**, and **Transpose**. Press **OK**.



All the average algorithm values will be pasted in one column.

	A	B	C
13	1.1150	1.1129	1.1205
14			
15			
16		1.11504	
17		1.11429	
18		1.1205	
19		1.12212	
20		1.12325	
21		1.12121	
22		1.12143	
23		1.12559	
24		1.12777	
25		1.1291	
26		1.12954	
27		1.13103	
28		1.13253	
29		1.13323	
30		1.13433	
31		1.13244	
32		1.13512	
33		1.13563	
34		1.13778	
35		1.13598	
36		1.13738	
37		1.141	

Select the depth values from the top row that contain the values of the depth soundings. Right click.

Select **Paste Special**Select the cell in **Column A** next to the top algorithm average value.

	A	B	C
13	1.1150	1.1129	1.1205
14			
15			
16	1	1.11504	
17	2	1.11429	
18	3	1.1205	
19	4	1.12212	
20	5	1.12325	
21	6	1.12121	
22	7	1.12143	
23	8	1.12559	
24	9	1.12777	
25	10	1.1291	
26	11	1.12954	
27	12	1.13103	
28	13	1.13253	
29	14	1.13323	
30	15	1.13433	
31	17	1.13244	
32	20	1.13512	
33	22	1.13563	
34	25	1.13778	
35	27	1.13598	
36	30	1.13738	
37	35	1.141	

Note: In some cases it is not possible to get at constant intervals. This is okay, but you might not be able to calculate the depth of extinction.

Note: If the chart sounding are in meters, then skip three steps forward.

In order to covert your depths from feet to meters (i.e., 1 ft is 0.305 m), select the cell in **Column c** next to the top algorithm average value and type **“=C16*0.305”** and press **Enter** on the keyboard.

Note: C16 is the location of the top value of the depth soundings in this example. Make sure to change the location to the top value location in your spread sheet.

14			
15			
16	1	1.11504	=A16*0.305
17	2	1.11429	

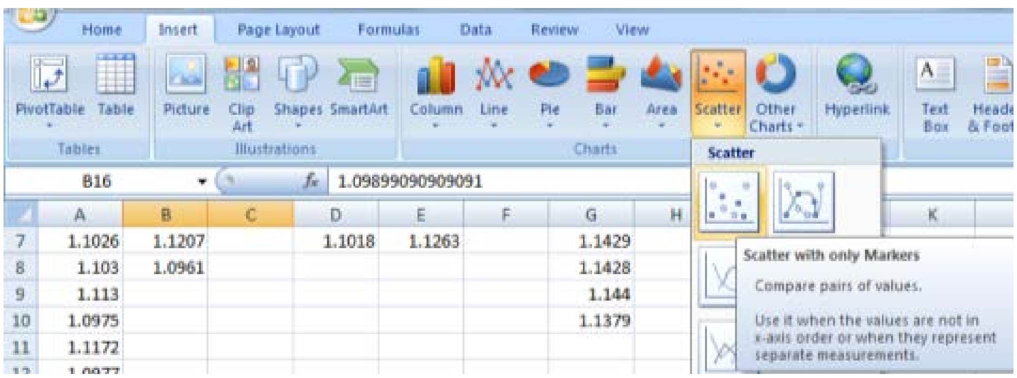
Repeat this step for the rest of the cells.

15			
16	1	1.11504	0.305
17	2	1.11429	0.610
18	3	1.1205	0.915
19	4	1.12212	1.220
20	5	1.12325	1.525
21	6	1.12121	1.830
22	7	1.12143	2.135
23	8	1.12559	2.440
24	9	1.12777	2.745
25	10	1.1291	3.050
26	11	1.12954	3.355
27	12	1.13103	3.660
28	13	1.13253	3.965
29	14	1.13323	4.270
30	15	1.13433	4.575
31	17	1.13244	5.185
32	20	1.13512	6.100
33	22	1.13563	6.710
34	25	1.13778	7.625
35	27	1.13598	8.235
36	30	1.13738	9.150
37	35	1.141	10.675

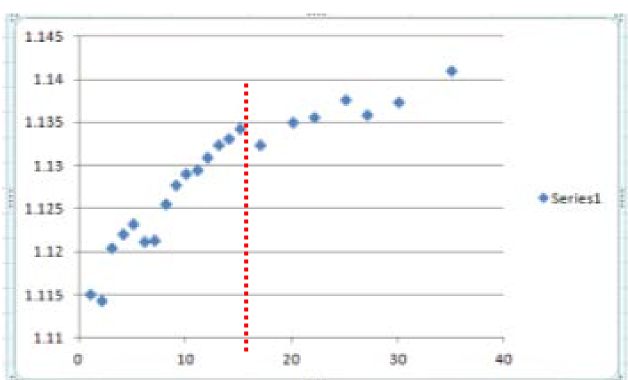
Select the values in **Column A** and **Column B**.

15			
16	1	1.11504	0.305
17	2	1.11429	0.610
18	3	1.1205	0.915
19	4	1.12212	1.220
20	5	1.12325	1.525
21	6	1.12121	1.830
22	7	1.12143	2.135
23	8	1.12559	2.440
24	9	1.12777	2.745
25	10	1.1291	3.050
26	11	1.12954	3.355
27	12	1.13103	3.660
28	13	1.13253	3.965
29	14	1.13323	4.270
30	15	1.13433	4.575
31	17	1.13244	5.185
32	20	1.13512	6.100
33	22	1.13563	6.710
34	25	1.13778	7.625
35	27	1.13598	8.235
36	30	1.13738	9.150
37	35	1.141	10.675

Select the **Insert** tab, then **Scatter**, and select the **Scatter with only Markers** graph choice.



The resulting graph will look similar to the following.



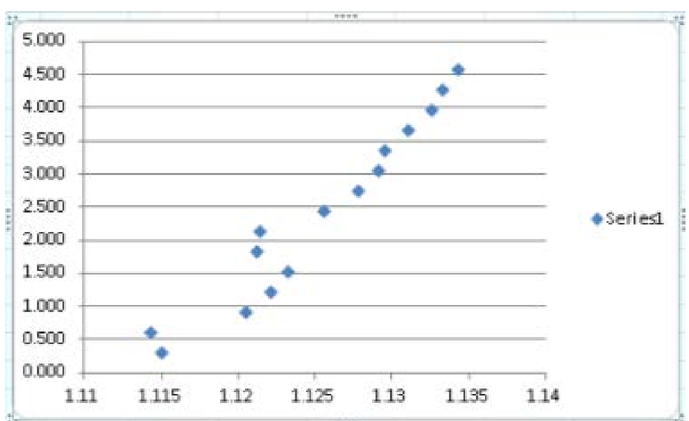
<u>Note:</u> Based on this plot we can see the depth of extinction is either 15 or 17 ft (around -5 m of the chart datum). Typically, beyond this depth there is a change in angle and/or there is less correlation between the sounding and the algorithm values.

Select the linear section of the plot (up to the depth of extinction) using **column B** and **column C**.

15			
16	1	1.11504	0.305
17	2	1.11429	0.610
18	3	1.1205	0.915
19	4	1.12212	1.220
20	5	1.12325	1.525
21	6	1.12121	1.830
22	7	1.12143	2.135
23	8	1.12559	2.440
24	9	1.12777	2.745
25	10	1.1291	3.050
26	11	1.12954	3.355
27	12	1.13103	3.660
28	13	1.13253	3.965
29	14	1.13323	4.270
30	15	1.13433	4.575
31	17	1.13244	5.185
32	20	1.13512	6.100
33	22	1.13563	6.710

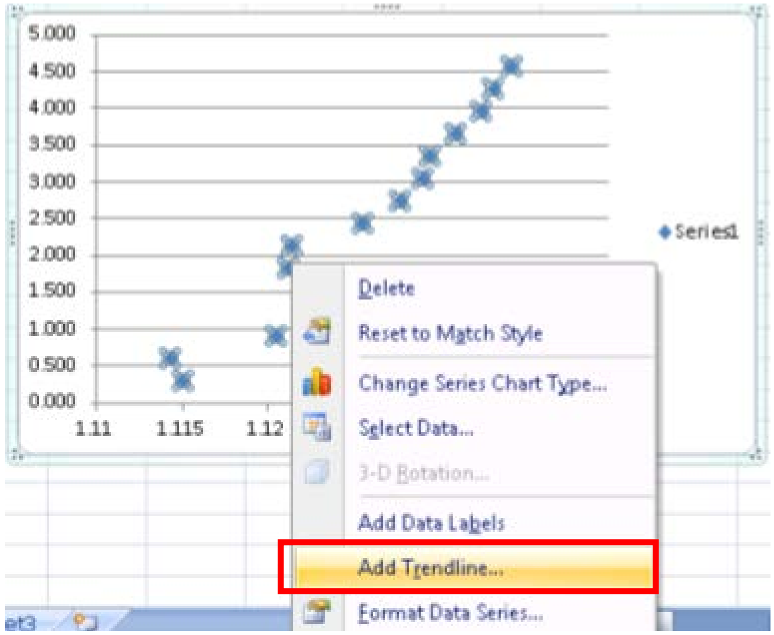
Select the **Insert** tab, then **Scatter**, and select the **Scatter with only Markers** graph choice.

The new plot will appear similar to the following.

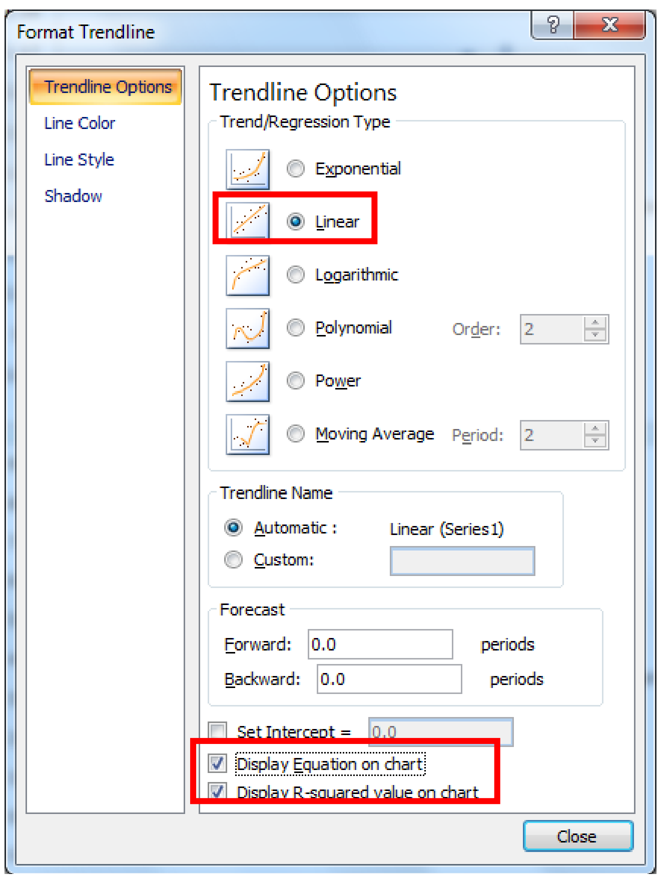


<u>Note:</u> Make sure that the algorithm results are presented in the X-axis.

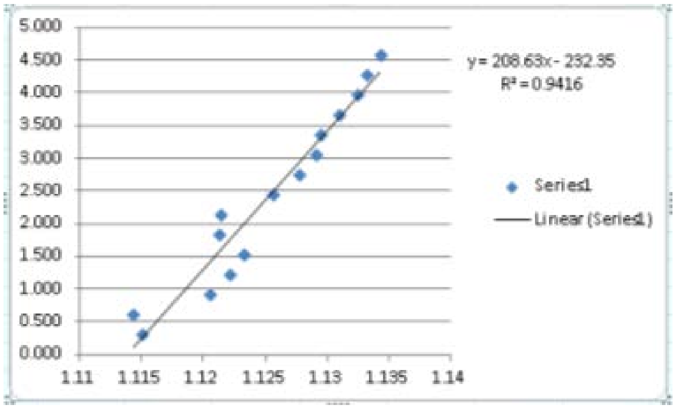
Right click the data points and select ***Add Trendline***.



In the **Format Trendline window** under the **Trendline Options** tab, select **Linear**, **Display Equation on Chart**, and **Display R-squared value**.

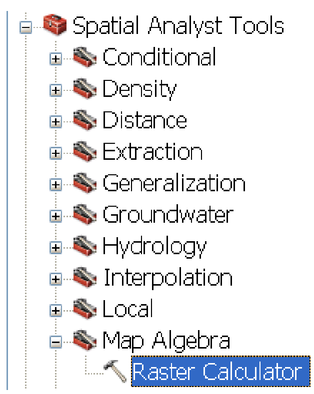


The chart now displays an equation that contains the gain and offset values that will be used to reference the algorithm result. Also, a correlation value (R^2) that evaluate the correlation between the scatter plot to a linear line (i.e., an ideal result will be very close is to 1.0).

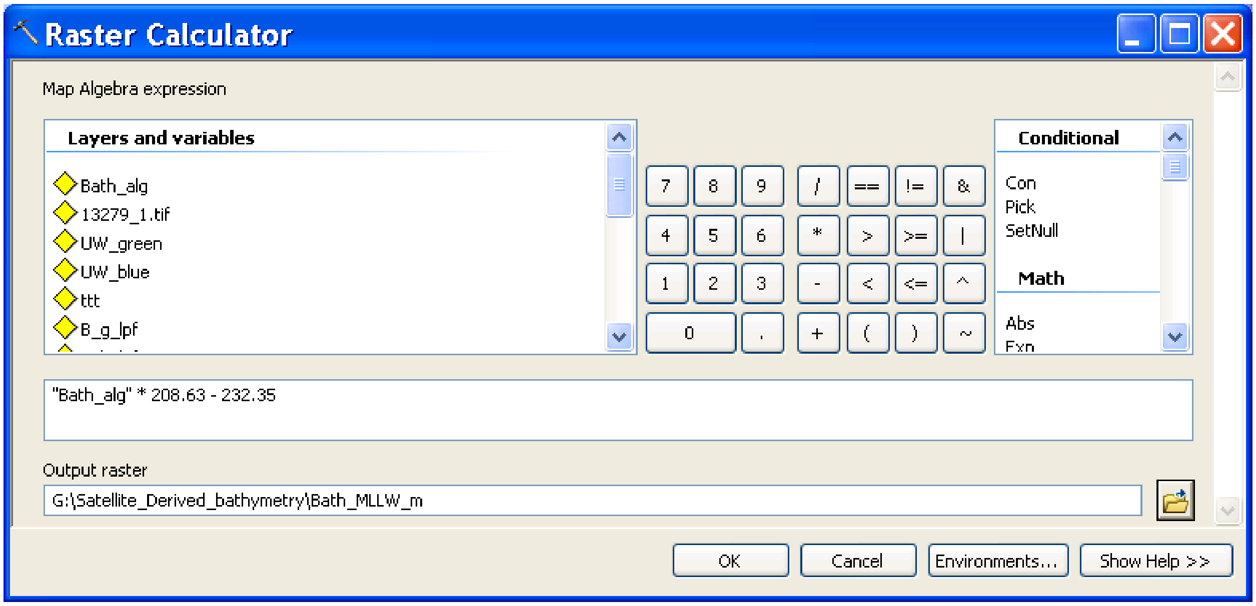


<u>Note:</u> The linear trend equation is in the form of y = ax + b, where a is the gain with **a** value of 208.63 and **b**is the offset with a valueof -232.35 to reference the algorithm result to the chart datum in meters.

Back inArcMap, select **Spatial Analyst Tools/Map Algebra/Raster Calculator** in the **ArcToolbox**.



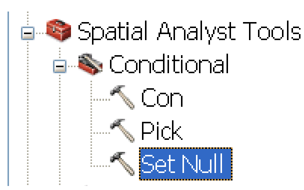
In the **Raster Calculator** window, type the following equation base on the Gain (**a**) and offset (**b**) values you calculated in MS Excel.Press **OK**.



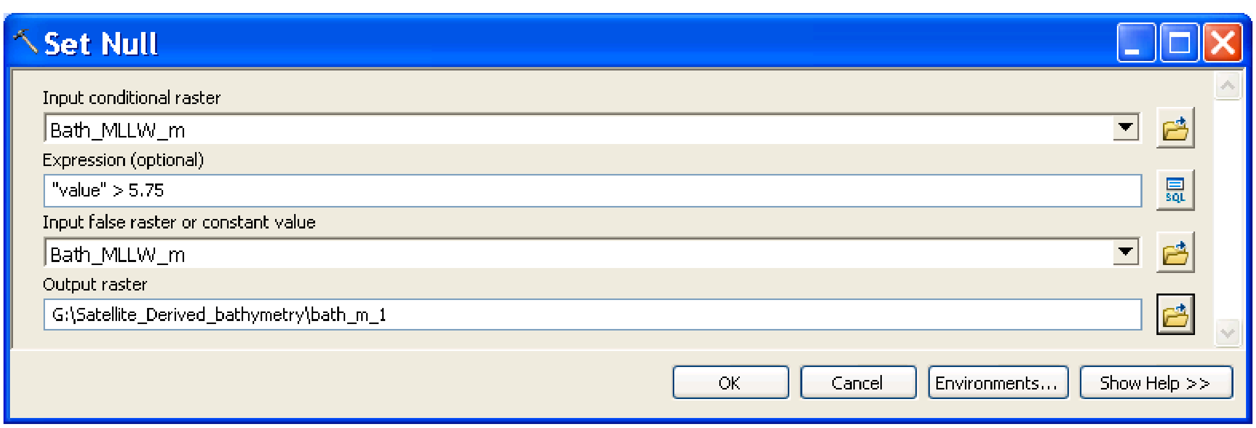
11.8 Post processing

11.8.1 Subsetting the LANDSAT 8 satellite-derived bathymetry

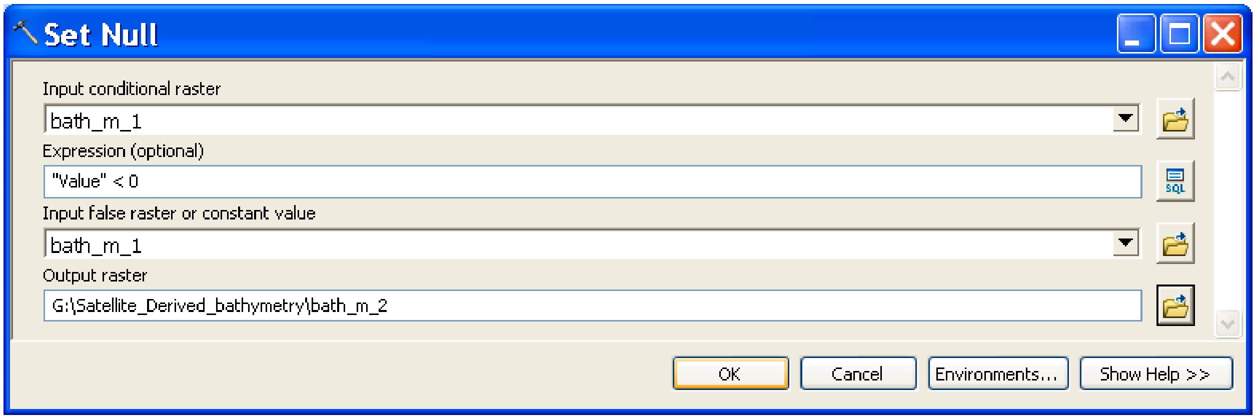
Remove the intertidal bathymetry and optically-deep bathymetry (deeper than the extinction depth), by selecting **Spatial analyst Tools/Conditional/Set Null**in the **Toolbox** window.



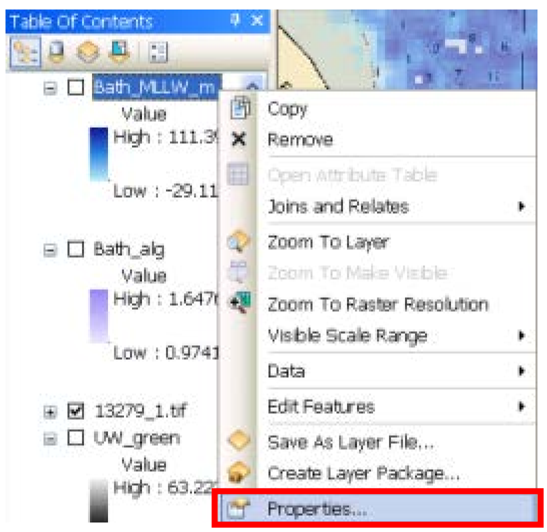
Fill the **Set Null** window as follows and press ***OK***. Make sure that the depth of extinction value measured from the MS Excel scatter plot (15 ft or 4.75 m) in the expression.



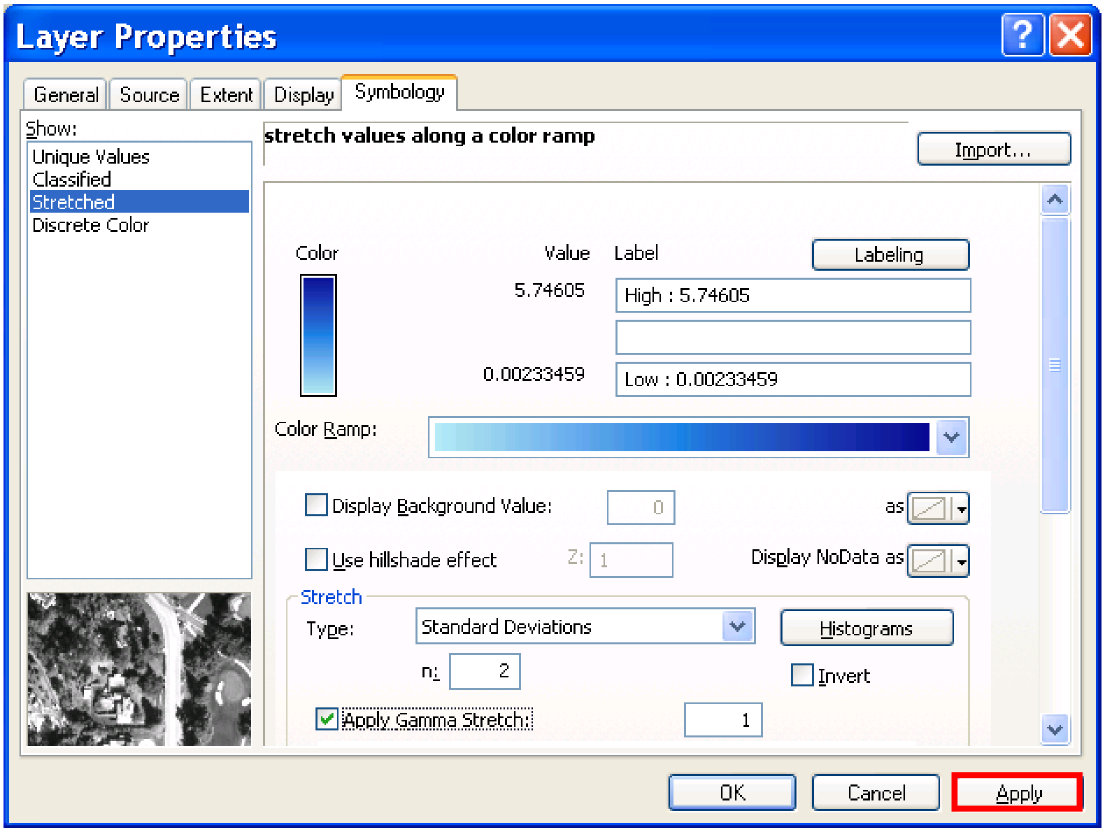
Similarly, select again **Spatial analyst Tools/Conditional/Set Null**from the **Toolbox** window and fill the following expression to remove the intertidal zone.



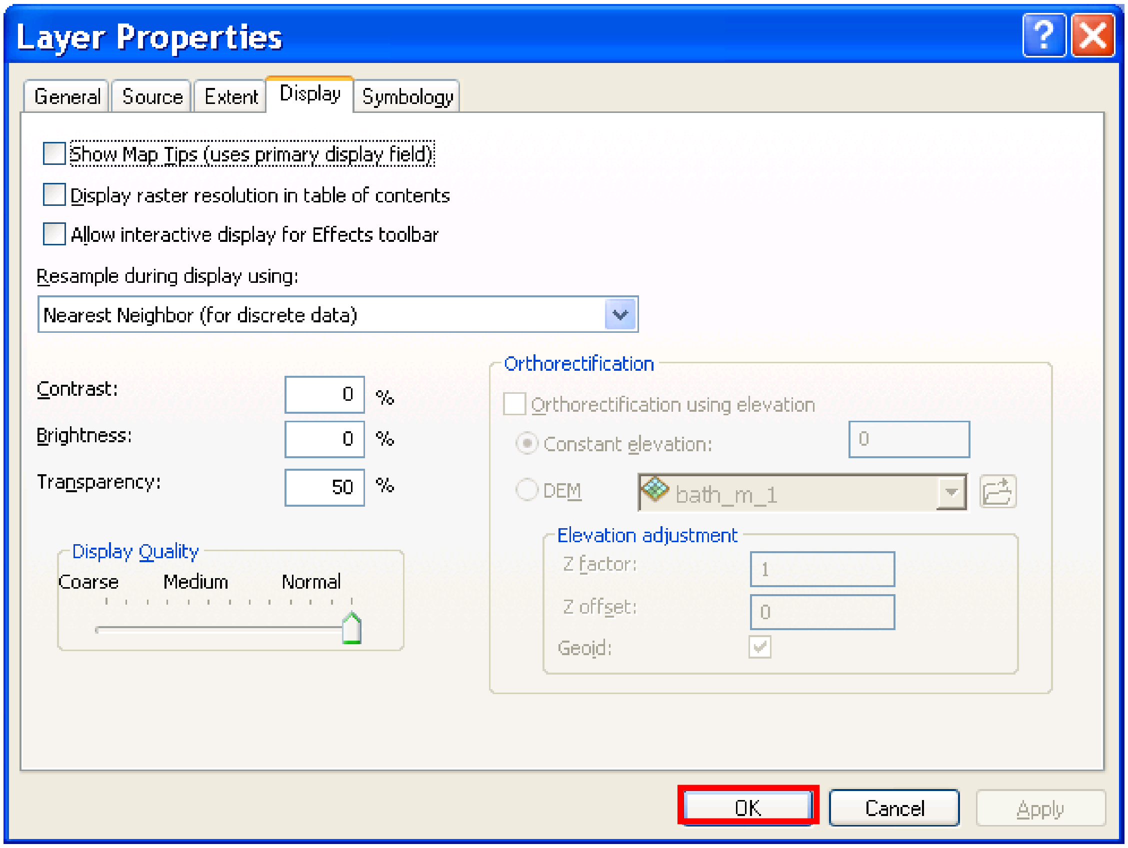
For visual inspection, zoom in to an area of interest (in shallow waters). Select the layer **properties**



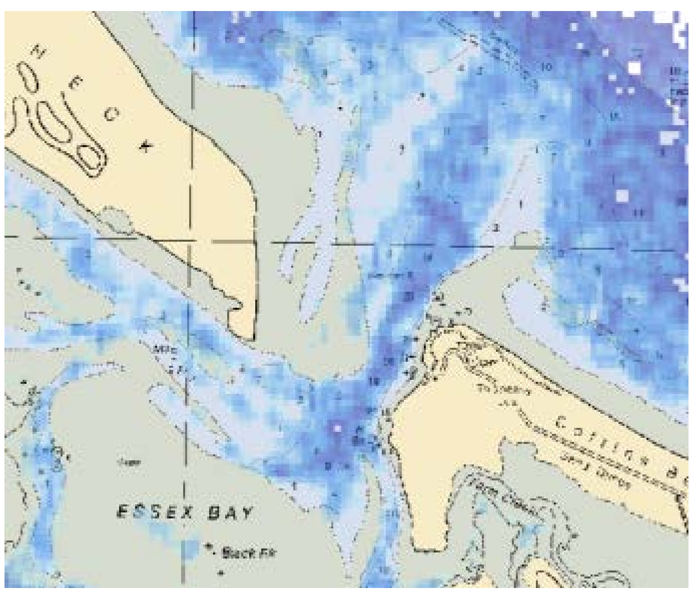
In the **Layer properties** window, select the **Symbology**tab and set the **color ramp and Apply Gamma Stretch: From Current Display Extent** as follows. Press **Apply.**



In the **Layer Properties** window, select the **Display** tab and set the**Transparency** to **50%**. Press OK.

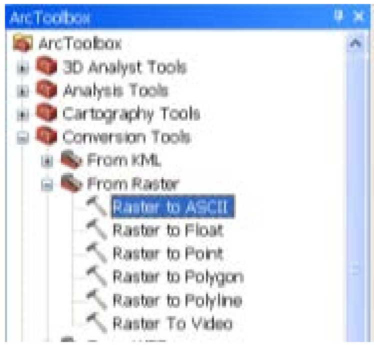


It is now possible to identify any changes between the satellite-derived bathymetry and the chart.

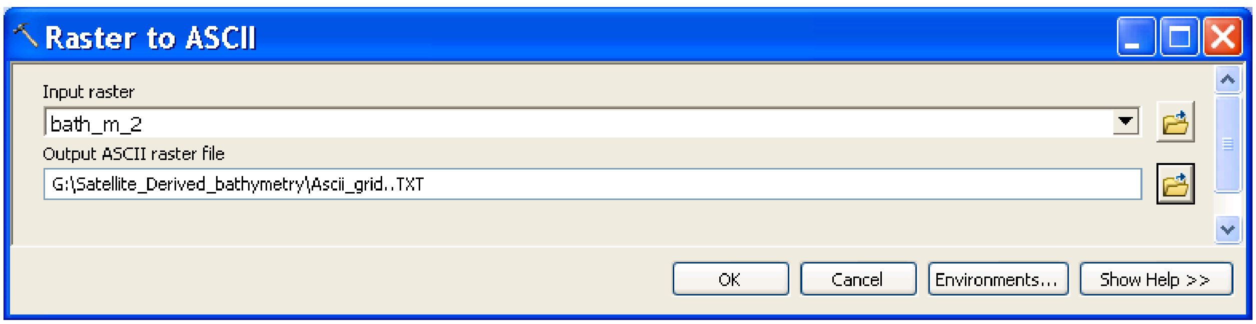


11.8.2 Exporting the bathymetry

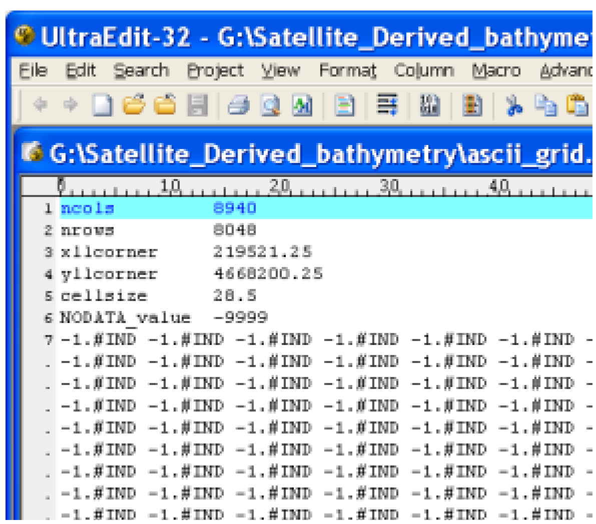
Select **Conversion Tools/From Raster/Raster to ASCII**from the **Toolbox** window.



In the **Raster to ASCII** window, select the clean bathymetry fileand select the name of the output file**.**



The outfile will be structure as an ASCII grid file

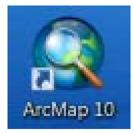


Appendix A – Referencing a scanned chart

Note: Make sure to acquire copyright permission from the hydrographic organization that produced the chart before scanning a chart.

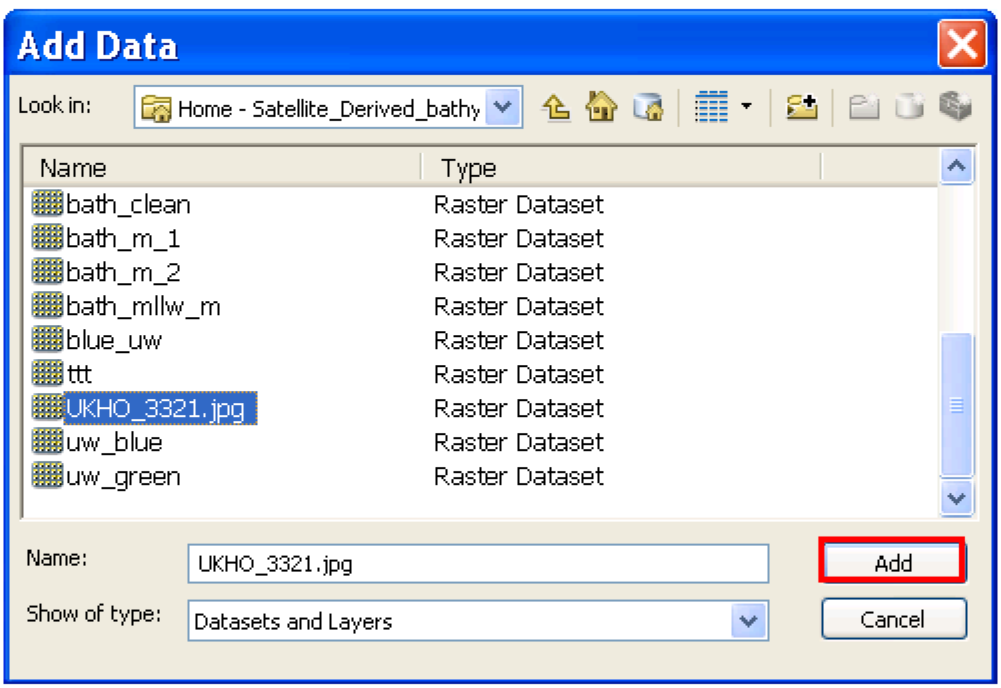
Scan your chart and save it to your computer

Open ArcMap.

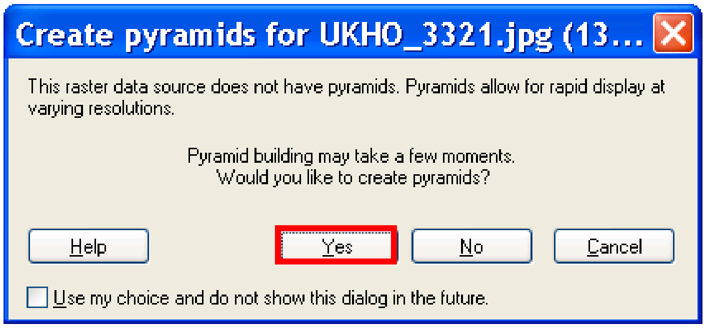


Load the scanned chart image to ArcMap by selecting the ***Add Data***…icon.

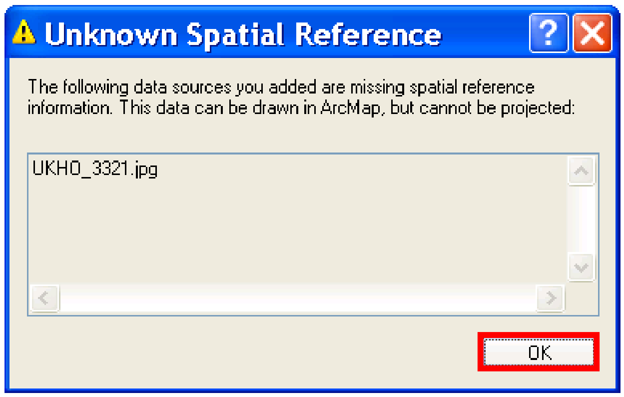
Navigate to your directory and select the chart in the **Add Data** window and select ***Add***button.



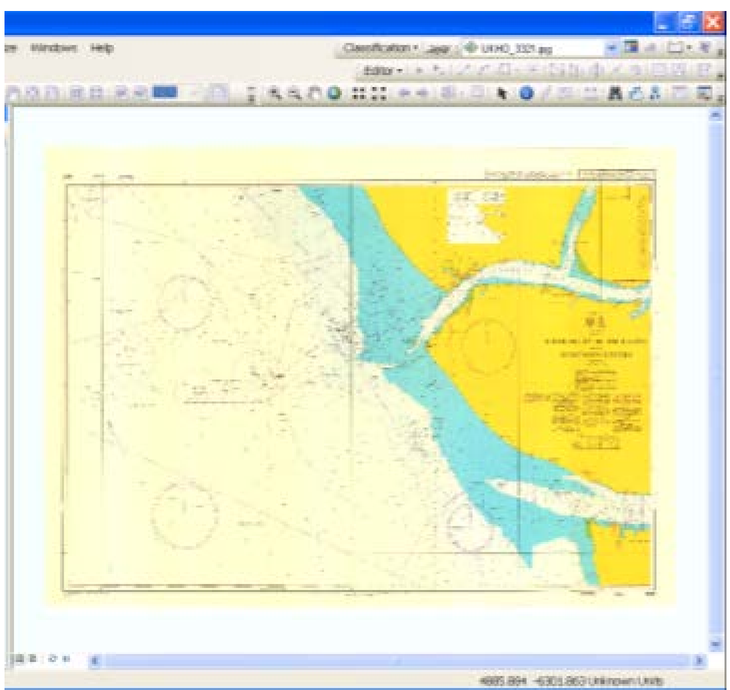
You will be asked to create pyramids. Press **Yes.**



You will also receive an**Unknown Spatial Reference** warning that this image is not referenced. Confirm this message by pressing on **OK**.

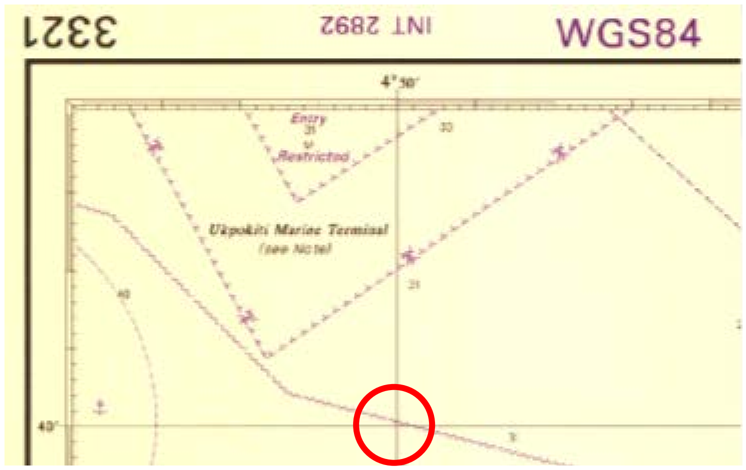


Your chart is now loaded into ArcMap

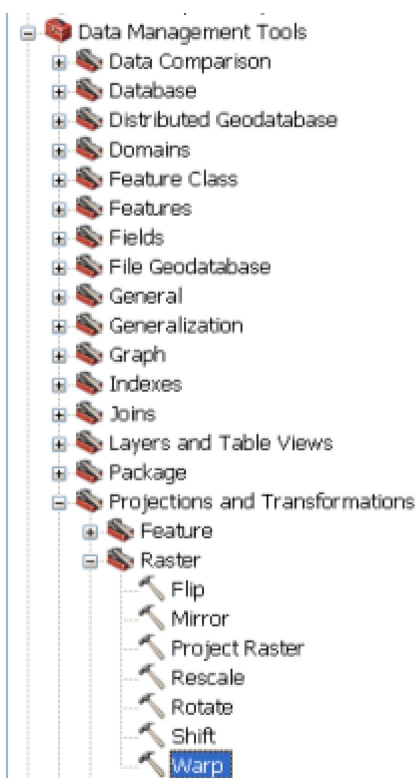


Select the **Zoom In**(magnify) icon

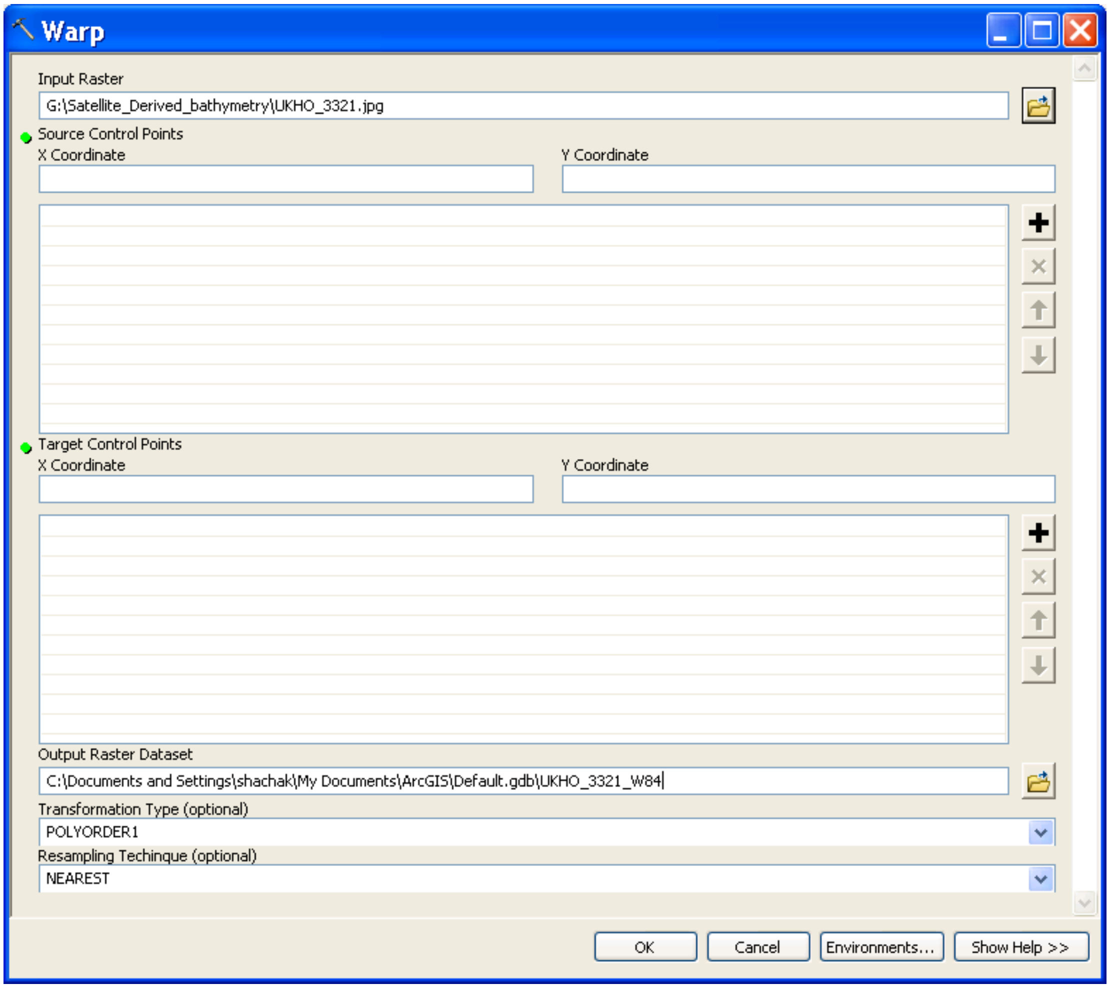
Zoom in to location that the longitude and latitude gratucules intersect



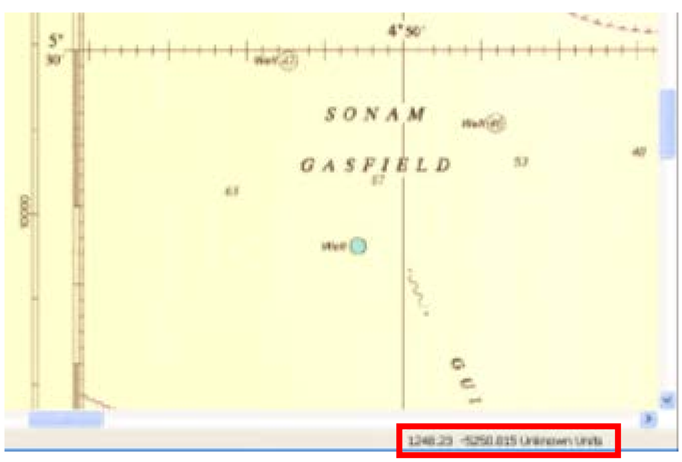
Select **Data Management Tools/Projections and Transformations/Raster/Warp**in the **Toolbox** window.



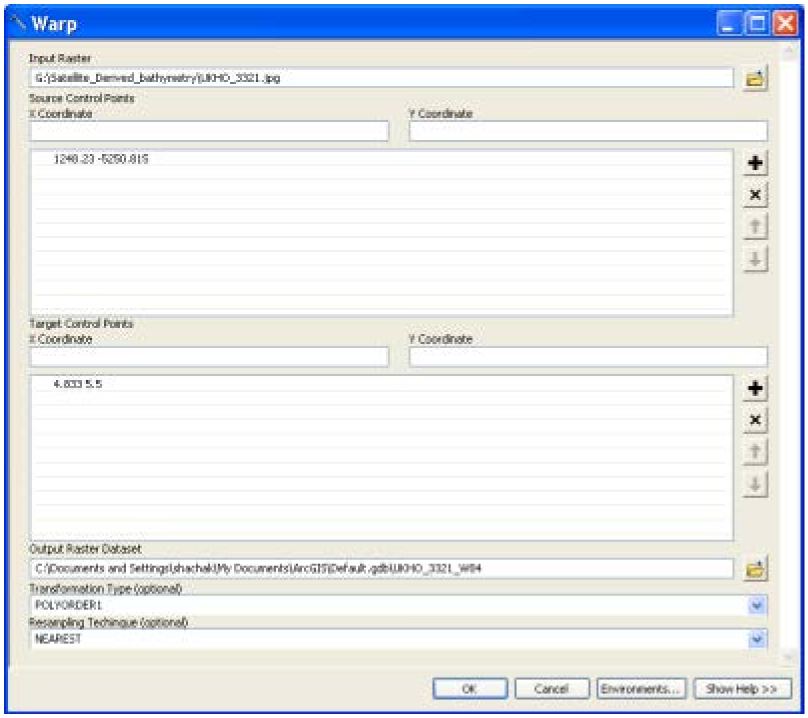
Load the input raster (i.e., the scanned chart) as **Input Raster** in the **Warp** window. Note the horizontal reference system used for the chart (in this example, WGS-84).



From the main view in ArcMap, start logging in the locations of the graticule intersection and the image position. In this example the source control points are X: and Y: and the Target Control points are X:4.833° and Y: 5.5° (long. 4°50' and lat.5°30').

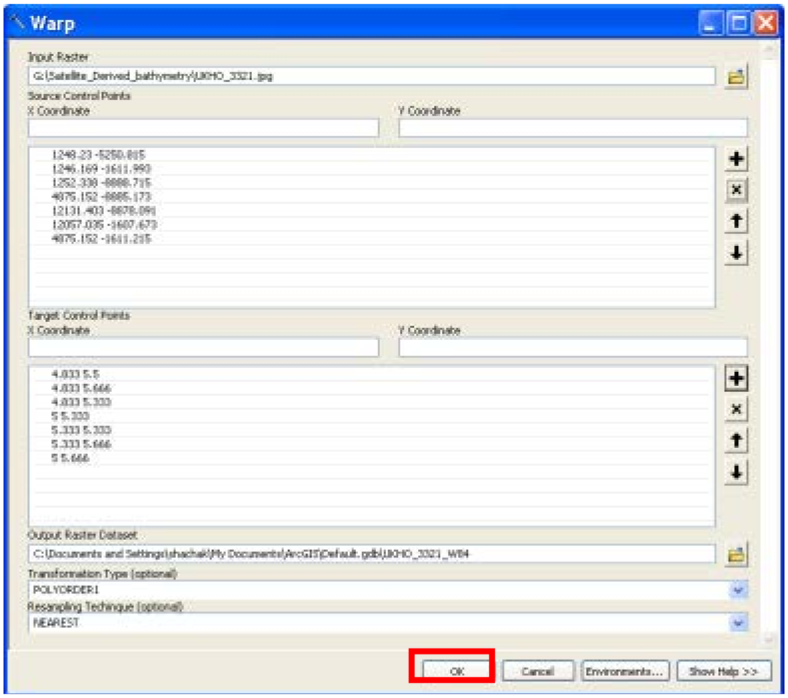


Type in the coordinates into the Warp window. Collect more points around the chart

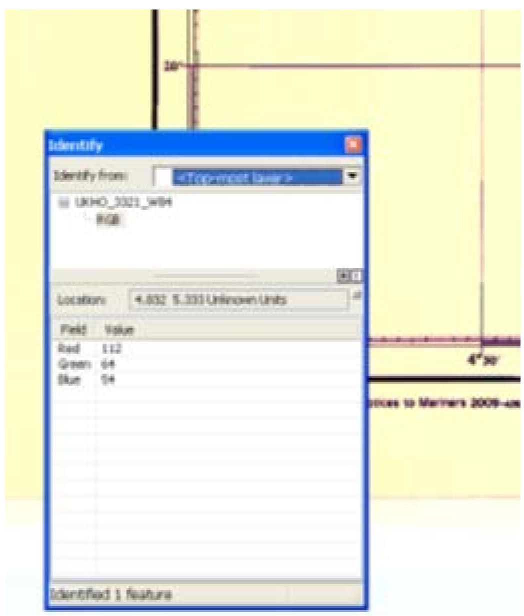


<u>**Note:**</u> Make sure to collect at least four points around the chart for a good referencing. Also make sure that the order of the target and source control points match.

After you have sampled the control points, make sure that the **Transformation Type: POLYORDER1** and press **OK** in the **Warp** window.

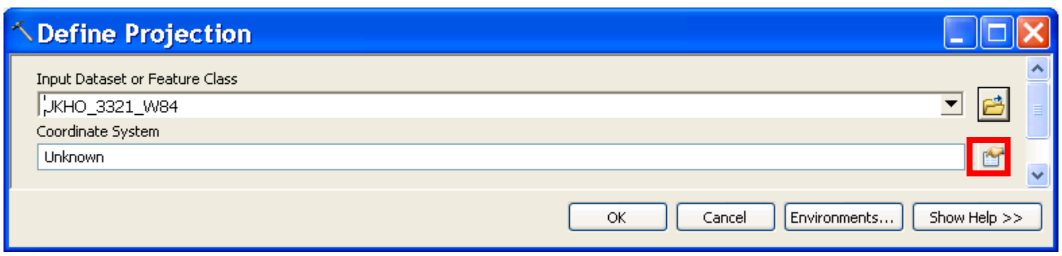


The warped image is now referenced to the horizontal reference system of the chart

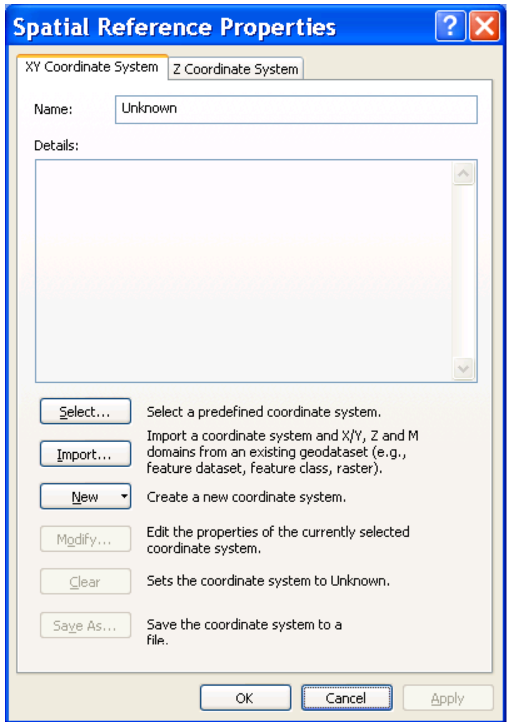


The last step is to define the reference system.Select **Data Management Tools/Projections and Transformations/Define Projection** in the **Toolbox** window.

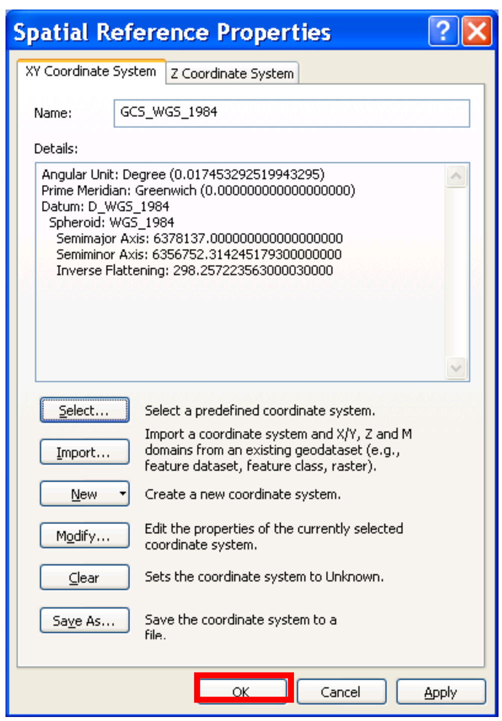
Load the warped image into the **Define Projection** window and press the **Select Reference Properties**.



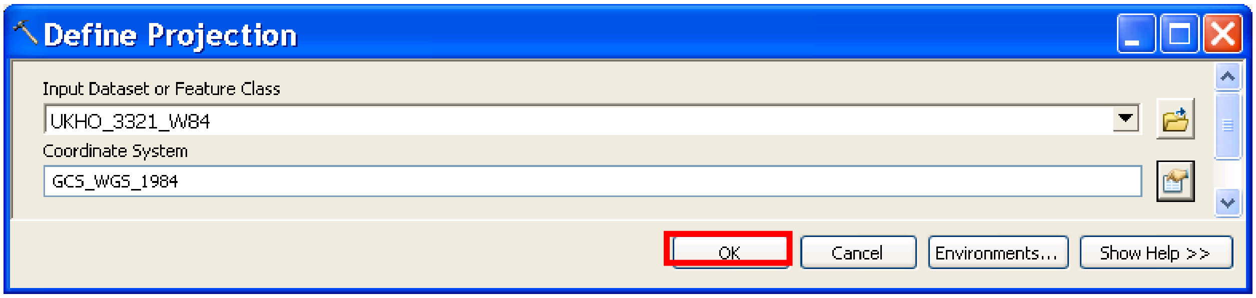
Press the **Select…** button in the **Select Reference Properties**window. In this example the reference system was **Geographic Coordinate System/World/WGS84.prj**.



After selecting a reference system, press **OK**.



Finally, press **OK** in the **Define Projection** window.


Define Projection
Input Dataset or Feature Class
UKHO_3321_W84
Coordinate System
GCS_WGS_1984
OK
Cancel
Environments...
Show Help >>

Annexes

Annex A Additional Resources

Multibeam Data

Baltic Sea Bathymetry Database
http://data.bshc.pro

Earthref.org Seamount Catalog
http://earthref.org/SC

European Marine Observation and Data Network
http://www.emodnet-hydrography.eu/

Geoscience Australia
http://www.ga.gov.au/index.html

University of Hawaii
http://www.soest.hawaii.edu/HMRG/Multibeam/index.php

Japan Agency for Marine-Earth Science and Technology (JAMSTEC):
http://www.godac.jamstec.go.jp/darwin/e

Lamont Doherty Earth Observatory Marine Geoscience Data System
http://www.marine-geo.org/portals/gmrt/

National Geophysical Data Center (NGDC)
http://www.ngdc.noaa.gov/mgg/bathymetry/multibeam.html

Center for Coastal & Ocean Mapping, Joint Hydrographic Center, University of New Hampshire
http://ccom.unh.edu/theme/law-sea

Submarine Arctic Science Program (SCICEX) data
http://nsidc.org/scicex/data_inventory.html

Stockholm University Oden Mapping Data Repository
http://oden.geo.su.se

Bathymetry Models

Altimetric Bathymetry Grid (Smith and Sandwell, 1997)
http://topex.ucsd.edu/WWW_html/mar_topo.html

GEBCO_08 Bathymetry Grid
http://www.gebco.net

International Bathymetric Chart of the Southern Ocean (IBCSO)
http://www.ibcso.org/data.html

International Bathymetric Chart of the Arctic Ocean (IBCAO)
http://www.ngdc.noaa.gov/mgg/bathymetry/arctic/

SRTM30_Plus
http://topex.ucsd.edu/WWW_html/srtm30_plus.html

Annex B Glossary

S-44 Terms

The following has been extracted from "IHO Publication S-44 Standards for Hydrographic Surveys" and edited for context.

Note: The terms defined below are those that are most relevant to this publication. A much larger selection of terms are defined in IHO Publication S-32 (Hydrographic Dictionary) and this should be consulted if the required term is not listed here. S-32 is available as an on-line Wiki at: http://hd.iho.int/en/index.php/Main_Page.

Accuracy: The extent to which a measured or enumerated value agrees with the assumed or accepted value (see: *uncertainty*, *error*).

Bathymetric Model: A digital representation of the topography (bathymetry) of the sea floor by coordinates and depths.

Blunder: The result of carelessness or a mistake; may be detected through repetition of the measurement.

Confidence interval: See *uncertainty*.

Confidence level: The probability that the true value of a measurement will lie within the specified *uncertainty* from the measured value. It must be noted that confidence levels (e.g. 95%) depend on the assumed statistical distribution of the data and are calculated differently for 1 Dimensional (1D) and 2 Dimensional (2D) quantities. In the context of this standard, which assumes Normal distribution of error, the 95% confidence level for 1D quantities (e.g. depth) is defined as 1.96 x sigma and the 95% confidence level for 2D quantities (e.g. position) is defined as 2.45 x sigma.

Correction: A quantity which is applied to an observation or function thereof, to diminish or minimize the effects of errors and obtain an improved value of the observation or function. It is also applied to reduce an observation to some arbitrary standard. The correction corresponding to a given error is of the same magnitude as the computed error but of opposite sign.

Error: The difference between an observed or computed value of a quantity and the true value of that quantity. (N.B. The true value can never be known, therefore the true error can never be known. It is legitimate to talk about error sources, but the values obtained from what has become known as an error budget, and from an analysis of residuals, are *uncertainty* estimates, not errors. See *uncertainty*.)

Metadata: Information describing characteristics of data, e.g. the *uncertainty* of survey data. ISO definition: Data (describing) about a data set and usage aspect of it. Metadata is data implicitly attached to a collection of data. Examples of metadata include overall quality, data set title, source, positional *uncertainty* and copyright.

Quality assurance: All those planned and systematic actions necessary to provide adequate confidence that a product or a service will satisfy given requirements for quality.

Quality control: All procedures which ensure that the product meets certain standards and specifications.

Reduced depths: Observed depths including all *corrections* related to the survey and post processing and reduction to the used vertical datum.

Sounding datum: The vertical datum to which the soundings on a hydrographic survey are reduced. Also called 'datum' for sounding reduction.

Total horizontal uncertainty (THU): The component of *total propagated uncertainty* (TPU) calculated in the horizontal plane. Although THU is quoted as a single figure, THU is a 2D quantity. The assumption has been made that the uncertainty is isotropic (i.e. there is negligible correlation between errors in latitude and longitude). This makes a Normal distribution circularly symmetric allowing a single number to describe the radial distribution of errors about the true value.

Total propagated uncertainty (TPU): the result of uncertainty propagation, when all contributing measurement uncertainties, both random and systematic, have been included in the propagation. Uncertainty propagation combines the effects of measurement uncertainties from several sources upon the uncertainties of derived or calculated parameters.

Total vertical uncertainty (TVU): The component of *total propagated uncertainty* (TPU) calculated in the vertical dimension. TVU is a 1D quantity.

Uncertainty: The interval (about a given value) that will contain the true value of the measurement at a specific *confidence level*. The *confidence level* of the interval and the assumed statistical distribution of errors must also be quoted. In the context of this standard the terms uncertainty and *confidence interval* are equivalent.

Uncertainty Surface: A model, typically grid based, which describes the depth uncertainty of the product of a survey over a contiguous area of the skin of the earth. The uncertainty surface should retain sufficient *metadata* to describe unambiguously the nature of the uncertainty being described.

Annex C Acronyms and Abbreviations

A

ASCII	American Standard Code for Information Interchange

B

BAG	Bathymetry Attributed Grid
BODC	British Oceanographic Data Centre (UK)

C

CHM	Center of Hydrography and Navigation
CHS	Canadian Hydrographic Service
CIOH	Centro de Investigáciones Oceanograficas e Hidrográficas (Columbia)
CIRES	Cooperative Institute for Research in Environmental Sciences (USA)

D

DEM	Digital Elevation Model
DHN	Directoria de Hidrografia e Navegaçao (Brazil)
DOC	Department of Commerce (USA)
DTM	Digital Topographic Model

E

F

G

GEBCO	General Bathymetric Chart of the Oceans
GEODAS	Geophysical Data System
GGC	GEBCO Guiding Committee
GIS	Geographic Information System
GMT	Generic Mapping Tools
GPS	Global Positioning System

H

HIPS	Hydrographic Information Processing System

I

IBCAO	International Bathymetric Chart of the Arctic Ocean
IHB	International Hydrographic Bureau (Monaco)
IHO	International Hydrographic Organization (Monaco)
IOC	Intergovernmental Oceanographic Commission

J

JAMSTEC	Japan Agency for Marine-Earth Science and Technology (Japan)
JODC	Japan Oceanographic Data Centre (Japan)

K

km	kilometer

L

LDEO	Lamont-Doherty Earth Observatory

M

m	meter
MBES	Multi-beam echo sounder

N

NERC	Natural Environment Research Council (UK)
NESDIS	National Environmental Satellite, Data, and Information Service
NGDC	National Geophysical Data Center (USA)
NOAA	National Oceanic and Atmospheric Administration (USA)
NOC	National Oceanography Centre (UK)
NRL	Naval Research Laboratory

O

OGC	Open Geospatial Consortium

P

Q

R

RAS	Russian Academy of Sciences (RUSSIA)
R/V	Research Vessel

S

SCICEX	Submarine Arctic Science Program
SHOM	Service Hydrographique et Oceanographique del la Marine (France)
SOEST	School of Ocean and Earth Science and Technology (USA)
SRTM	Shuttle Radar Topography Mission

T

TAR	Tape Archive software utility
TIN	Triangular Irregular Networks

U

UNH	University of New Hampshire (USA)
URL	Universal Resource Locator
UTM	Universal Transverse Mercator

V

W

WWW	World Wide Web

X

Y

Z